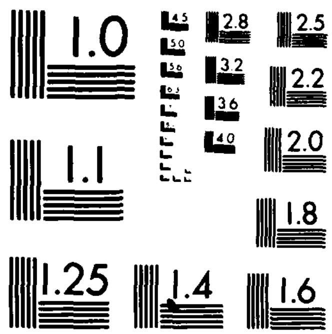
AD-A172 448 INVESTIGATION AND DESIGN OF A PROJECT MANAGEMENT DECISION SUPPORT SYSTEM F..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. 1/2

UNCLASSIFIED R H BLACK ET AL. MAR 86 AFIT/GST/ENS/86M-2 F/G 5/1 NL




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AFIT/GST/ENS/86M-2



INVESTIGATION AND DESIGN OF A
PROJECT MANAGEMENT DECISION SUPPORT SYSTEM
FOR THE 4950TH TEST WING

THESIS

Robert H. Black
Major, USAF

Mark J. Fowler
Captain, USAF

AFIT/GST/ENS/86M-2





Approved for public release; distribution unlimited

AFIT/GST/ENS/86M-2



# INVESTIGATION AND DESIGN OF A PROJECT MANAGEMENT DECISION SUPPORT SYSTEM FOR THE 4950TH TEST WING

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Operations Research

Robert H. Black
Major, USAF

Mark J. Fowler
Captain, USAF

March 1986

Approved for public release; distribution unlimited

## PREFACE

The purpose of our study was to investigate methods for putting a large central data base to use for the 4950th Test Wing, Wright-Patterson AFB, Ohio. From the beginning, our research centered on using the data base to aid in project scheduling. The initial concept aimed toward developing a scheduling algorithm which could incorporate several unique elements of project flow within the wing. Early on, however, we found that the real problem facing the wing was in breaking out of old ways of looking at problems; hence, our study became one of developing a new problem solving method: one adept at solving fluid problems with indeterminant quantities, one flexible enough to handle daily changes without overtaxing the users, and most importantly one concentrating on the needs of the commander in making wing decisions.

Earlier research into project management and scheduling problems has largely centered on techniques for generating schedules, not on aiding decision makers in comparing the schedules to determine which they wish to implement. In contrast, our study develops a problem solving methodology which begins at the end - with the decision maker - and works backwards to determine system requirements. In a classic systems analysis approach, we selected a small, potentially solvable problem: how to select the best wing schedule. Understanding that "best" must be defined in terms of meaningful organizational goals, we then sought out the goals of the organization, the measureable objectives they wish to meet with regard to scheduling, and the project variables the wing can manipulate to effect changes in the flow of projects and hence changes to the schedule. At this point, we broke with traditional operations research techniques by not trying to specifically determine the optimum schedule. Rather, we concentrated on the needs of the decision maker - how the decision maker might view the scheduling problem and what information the decision maker needs to see in order to make a decision. We maintain that all other system requirements (OR/MS models and algorithms, simulations, raw data form and content) flow from these decision maker needs and not vice versa.

In our study, we design a decision support system for wing project management and scheduling. In the design, we make heavy use of the ROMC approach presented by Ralph Sprague and Eric Carlson. Our effort is not, however, a mere implementation of a previously developed methodology. We advance the need for maintaining a constant view to the end user (the decision maker) and his requirements for making decisions. The presentation of essential information in a manner which allows the decision maker's mental decision making process to flow unimpeded is presented as the central concern for developing useful information systems.

In reviewing current information system literature, we found contributors did not commonly approach problem solving from the decision maker's view, but from the opposite end - beginning with a problem, searching for a technique to generate solutions, then finally realizing that someone must use the information to make a decision. The result was often forcing the user to live with the generated output, instead of forcing the output to meet the needs of the decision maker. Our study identifies several possible reasons for this mismatch between services and requirements and provides recommenations for their minimization.

In sum, we believe the our methodology and findings can significantly aid organizations in building systems to support decision making. Such systems are becoming more important to decision makers as increased emphasis is being placed on solving large, difficult to define problems involving complex internal interactions in a rapidly changing environment (for example, the command and control of military forces during a crisis). With the increasing availability of advanced graphics, modeling, and data base systems for use on microcomputers, our methodology provides opportunities for improvements in decision making at all levels.

Having finally extolled our virtues to the Air Force, we now wish to formally acknowledge the efforts of several people without whose aid and support we could not possibly have completed our endeavors. First, we must thank the personnel of the 4950th Test Wing, and in particular Lt Col Don Sutton, for allowing us to intrude in their problems. We sincerely hope our meager efforts will help them put their WIS to the best use possible. Secondly, we extend our thanks to Lt Col John Dumond in the AFIT School of Systems and Logistics whose assistance expanded our view of the problems associated with the computerized generation of project schedules.

Certainly, no thesis effort could be adequately completed without the active involvement of a knowledgable advisor. Major Skip Valusek must be credited with our initial exposure to the concept of decision support systems, with healthy doses of unobtrusive guidance throughout our research, and with the courage to allow us to flail about while expanding the current frontier of decision support concepts.

Finally, and certainly mostly, we acknowledge the crucial support provided us by our families. Ainslie, Cathy, and all the munchkins have shown a nearly endless flow of patience and understanding - perhaps more than we deserve. They unquestionably share in our degrees.

Robert E. Black
Mark J.Fowler

## Table of Contents

# List of Figures

## List of Tables

## Abstact

This study investigated methods for putting a large central Management Information System (MIS) data base to use for the 4950th Test Wing, Wright-Patterson AFB, Ohio. The study focused on using information to help the command section make decisions regarding project scheduling and management. Within an overall framework of systems analysis, this study used the Representations, Operations, Memory Aids, Control Mechanisms (ROMC) approach developed by Ralph Sprague and Eric Carlson to design a Decision Support System (DSS) for the test wing. This study advances DSS design theory in showing the overriding importance of the decision maker and his needs in defining DSS requirements. The general observations of this study, along with the advances in design methodology can significantly aid organizations in building systems to support decision making. Such systems are becoming more important to decision makers as increased emphasis is being placed on solving large, difficult to define problems involving complex internal interactions in a rapidly changing environment (as in the command and control of military forces during a crisis). With the increasing availability of advanced graphics, modeling, and data base systems for use on microcomputers, the methodology presented here provides opportunities for improvements at all levels.

INVESTIGATION AND DESIGN
OF A
PROJECT MANAGEMENT DECISION SUPPORT SYSTEM
FOR THE
4950th TEST WING

# I. Introduction

## Purpose

The purpose of this effort is to investigate the project management decision making process of the 4950th Test Wing, Wright-Patterson AFB, Ohio, and to help design a system to aid the commander make project scheduling and management decisions. Traditional operations research and management science approaches to such project management tasks tend to rely on scheduling algorithms and optimization schemes that produce a single "best" schedule. While these procedures can be beneficial in cases where the decision to be made remains constant, they tend to neglect the decision maker and the decision making process by focusing only on one pre-specified set of goals and constraints: they assume conditions will never change and, frequently, take the authority for making decisions away from the responsible decision maker. This effort focuses on the project management decision making process and the need for assisting the decision maker by allowing the comparison of alternatives. It recognizes the importance of easily understood and workable heuristics, compromise between conflicting organizational goals, and the ability to change views and goals without trying to model each one explicitly in a specific, inflexible optimization algorithm.

To begin the investigation, Chapter I examines the current scheduling procedures within the 4950th Test Wing, leading to a concise statement of the specific problem to be addressed. In Chapter II, the study investigates several project management and scheduling methodologies from the view of how they might support the decision making process. This investigation leads to the selection of the Decision Support Systems (DSS) approach for this effort. Chapter III discusses DSS in general and includes explanations of how traditional optimization methodologies can be incorporated into a flexible DSS package to aid the decision maker. The development of the specific DSS for project management and scheduling in the 4950th Test Wing is examined in detail in Chapter IV and is followed by recommendations for implementation and evaluation in Chapter V. Chapter VI concludes with general observations about decision support systems in the military.

## Wing Background

The 4950th Test Wing, Wright-Patterson AFB, Ohio, provides test support to the divisions of the Air Force Systems Command and other Department of Defense agencies involved in research and development for the armed forces. A major portion of the work performed by the wing is flight test and evaluation of aircraft electronics components. Thus, the wing is project oriented; that is, the wing's schedule, work load, manning, and resources are all driven by the projects they handle.

The wing may work on as many as 200 projects at any given time. Each project makes varying demands on the resources of the wing. While individual projects are unique in the specific demands placed on the functional divisions of the wing, a typical project will require several distinct steps. The test director and the Test Engineering Branch must design the test to perform, while the Aircraft Modification Center designs a way to mount the test equipment in the aircraft, procures the materials required, modifies the aircraft, and installs the test equipment. After the airborne tests, the Data Analysis Section of the Test Engineering Branch must evaluate the test results, while the Modification Center removes the equipment and returns the aircraft to its previous condition. The test team then prepares a formal report on the project. These steps are not necessarily sequential; but, many steps depend upon previous steps for their completion, as illustrated in Figures 1.1 and 1.2. These interrelations necessitate accurate, yet flexible, scheduling.

Because of limited resources, the commander, through the test director, must decide how best to fit projects into the wing schedule, while meeting due date requirements set by customers. This requirement applies to changed or delayed projects as well as new requests for wing services. The test director must determine the impact each project will have on current projects and resources and then forecast a

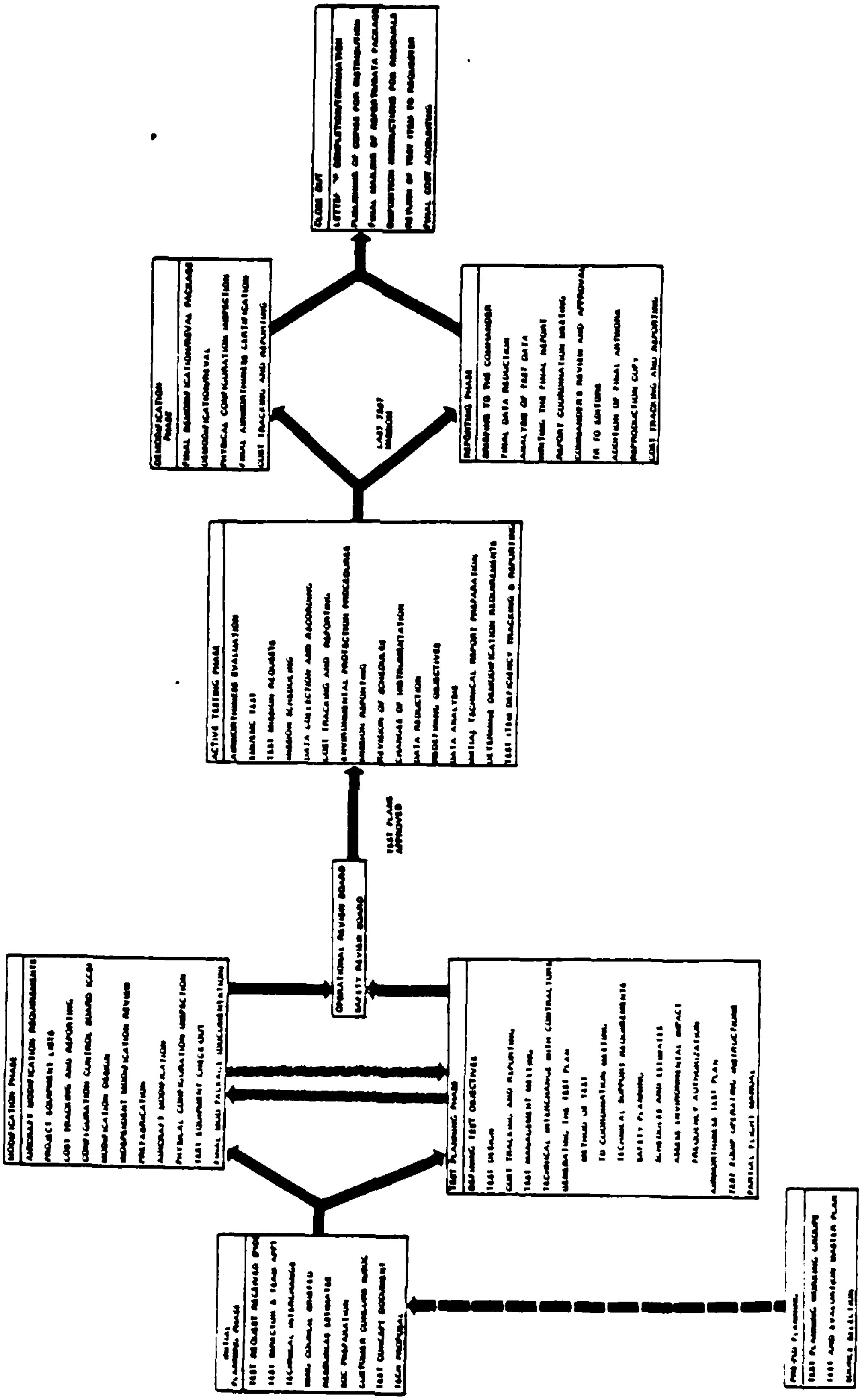


Figure 1.1 Typical Project Flow Through the 4950th Test Wing
(Test Guide, 1983: 1-9)

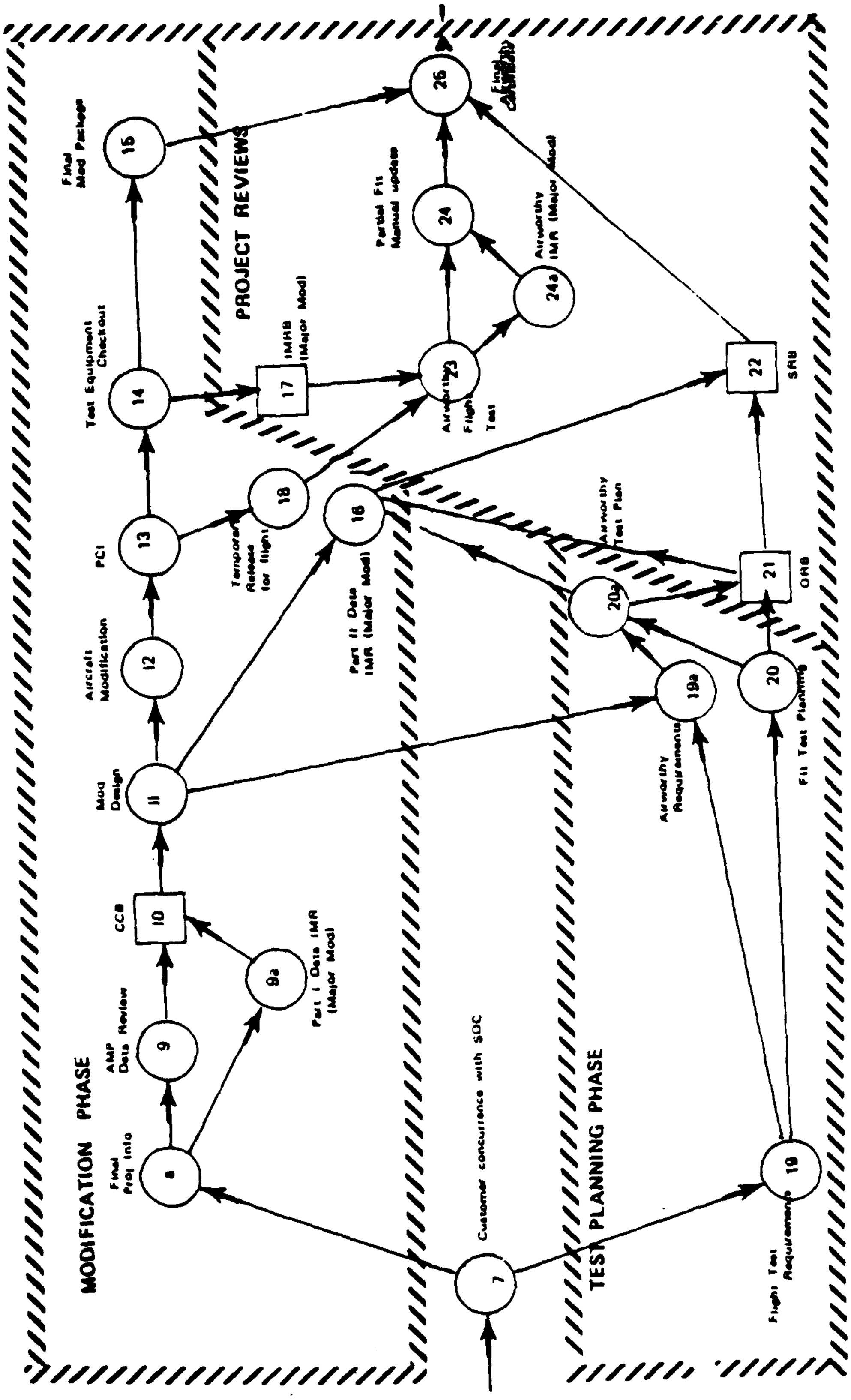



Figure 1.2 Typical Project Flow Through Modification, Test Planning and Review Phases (Test Guide, 1983: 4-19)

schedule to minimize those impacts. To judge these impacts and make scheduling recommendations to the commander, the test director requires accurate information about the state of the wing, current and projected, and a method for testing the impact of new and changed projects. As discussed below, current scheduling procedures do not satisfy this need.

## Current Procedures

Currently, as each new request for wing services arrives, the commander appoints a test director and test team composed of at least one person from each functional area involved in the project. The team must try to identify the resources, manpower, and time requirements of the project. Shop chiefs provide estimates of the impact each project will have on their individual shops (Test Guide, 1983). The test director consolidates the inputs from the shops and manually determines how each project will be scheduled. The test director must use heuristic scheduling judgement to fit new projects into the existing schedule: judgement that is based on unstructured and project-unique ingredients including relative priorities, flexibility of tasks, timing of due dates, and allowable variability in test objectives. While the end product is known to be a schedule, the process of arriving at a complete schedule involves a series of qualitative judgements which cannot be adequately or accurately automated.

Under the current system, the commander and test directors have no analytical capability to view the interactions of projects and shops within the wing. With the exception of the modification center, shop chief estimates are made in isolation: the estimated time to complete a project is based on the complexity of the project alone without consideration of the impacts of previously scheduled projects competing for the same limited resources. In practice, the modification center provides a time window to the Test Director. The other shops appear to ignore the project until it arrives for their work, whereupon they work as best they can to meet any due date constraints (Interviews, 1985). The results can induce large fluctuations in manpower usage, compromises in testing quality for lack of time, and late completions. Essentially, the schedule is forced without direct involvement of the decision making authorities in the wing and without reference to any overall wing goals or objectives.

Additionally, the wing has no day-to-day capability to investigate the effects of project deviations. If a project progresses slowly in one shop or requires additional unplanned resources, the shop may not be able to keep its other projects on schedule. Likewise, a late project may affect the schedules of other shops as the project flows through the wing. Changes in any project might affect the capabilities of the wing, yet the wing has no capability to test effects before changes are made nor to efficiently notify shops and test teams affected.

Less obvious deficiencies in the current system concern the ability to integrate knowledge from the several diverse areas of the wing and the loss of knowledge and wisdom the wing experiences with personnel changes. Since project resource and requirement estimates are made with heuristic rules developed through each shop chief's experiences with past projects (Interviews, 1985), if a shop chief or staff member leaves, the heuristics are lost; a new person in the job may not have the benefit of experience for estimating project requirements. The commander cannot, then, be assured of accurate information and a wise decision.

## Problem Environment

Wing Recognition of Deficiencies. In March of 1984 the 4950th Test Wing commander ordered a review of the wing mission. The purpose of the review was to "identify problems in fulfilling the mission, and propose information systems which would support the many decisions facing the Wing each day" (Glenn, 1984:1). The wing used the Business Systems Planning (BSP) methodology as developed by the International Business Machine (IBM) Corporation. The review identified the steps involved in providing test support, classes of information required to perform those steps, and where each item of infoımation required was created and used. The review then identified 26 problem areas related to the flow of information within the wing. The most important problem was Tactical Planning: accurate scheduling, tracking of project changes and their impacts on other projects, and testing the results of accepting new projects into the wing schedule (Glenn, 1984). As a result of the BSP study, the commander committed the wing to a long term program of integrating all wing information needs into an overall management information system designed to facilitate collection of data and generation of routine reports. The BSP study did not, however, concern itself with the end use of information flowing through the organization; particularly, not with the effective presentation and use of the information for decision making.

Wing Goals and Objectives Relating to Project Management and Scheduling Decisions. For the commander to choose between several potential schedules, he must somehow measure how well each schedule meets the goals of the organization. The wing considers efficient budget allocation and customer satisfaction as their major goals and measures of success in project management and scheduling (Sutton, 1985). Each of these goals is examined to find quantifiable measures of performance, or objectives, to aid in the comparison of schedules.

Budget Allocation. The wing has two types of budgeting authority that must be properly balanced to ensure the wing can perform its mission. Direct Budget Authority is designed to pay for all wing activities not directly related to a project. This 40 percent of the annual wing budget of approximately $90 million must cover aircrew flight training, civilian pay for hours not spent on reimbursable projects, and "maintaining (and modernizing) test capabilities" (Glenn, 1984:10). The other 60 percent of the wing budget falls under Reimbursement Budget Authority. All costs directly related to a specific project are tracked and charged to the customer. The challenge to the budget is summed up in a recent Wing Business Systems Planning study: "Since changes to the Wing's basic test resources . . . cannot be made quickly, a reduction in workload (reimbursable funding) must be offset by . . . additional institutional [direct] funding" (Glenn, 1984:11). Thus, the major objectives for the wing budget involve accurate scheduling and workload forecasting to allow acceptance and completion of as many projects as possible to keep wing personnel gainfully employed with reimbursable projects, thereby reducing the impact on the wing's direct budget, without unnecessary overtime work

(Sutton, 1985). This point also impinges directly on customer satisfaction.

Customer Satisfaction. While organizations requesting wing flight tests are concerned about costs, two other factors also directly affect the customer's satisfaction with the test wing: quality of testing and test completion dates (Sutton, 1985). Quality of testing involves performing the tests properly: gathering and analysing the correct data for the customer. Testing requires time, and time implies a dependence on the wing schedule. Test completion dates are critical to customer satisfaction since, in general, the customer requires results by specified dates or the usefulness of the tests may be lost (Interviews, 1985). Ensuring project completion by the required due dates depends on the accuracy of the wing schedule, changes to the schedule, and decisions to accept new projects into the wing schedule. As discussed earlier, any change in the progress of one project in one shop affects the schedule of other projects and the capabilities of the entire wing. Likewise, decisions to accept new projects may induce changes in the wing schedule and cause the forecast completion dates of other projects to change. The major objectives of the wing with respect to customer satisfaction, then, are minimizing project completion delays and cost overruns while meeting customer requirements for quality testing. This should be accomplished through accurate tracking of changes induced by delays and forecasting the impacts of new projects to aid the commander in making more informed decisions (Sutton, 1985).

Summary of Objectives. As presented above, the wing has four measureable objectives supporting their scheduling goals in the areas of budget allocation and customer satisfaction:

1) Complete as many reimbursable projects as possible.
2) Minimize overtime and other cost overruns.
3) Minimize due date delays.
4) Maximize the quality of testing.

Operative Variables and the Generation of Alternative Schedules. Making a decision between alternative schedules based on their relative attainment of organizational goals presupposes the existence of multiple schedules with different impacts on those goals. Such alternative schedules can be generated by varying the operative variables of the organization in the areas of project management and scheduling. Operative variables are those conditions over which the commander has control: those actions the commander can take to effect some change in the problem situation. Once the variables are identified, they can be systematically changed to determine their effects toward meeting the goals of the organization. Such tests with variations in conditions allow the commander to effectively choose the course of action (schedule) best meeting the goals of the organization. The variables over which the 4950th Test Wing has control include:

1) Work capacity
2) Completion dates
3) Priorities of projects
4) Performance
5) Modification procedures
6) Aircraft utilization

(Sutton, 1985; Interviews, 1985).

The commander may vary the work capacity of the organization by requiring weekend flight testing, longer duty days for military personnel, or overtime work for civilian workers. He may allow slippage of completion dates or change project priorities to allow some projects to move ahead of others. Additionally, he may reduce the scope of a test to allow for faster completion. If he finds problems in the modification of aircraft, he may be able to shift modification responsibilties to the AMX shop, or he may authorize contracting of the modification to a civilian corporation. The commander may also authorize placement of several projects simultaneously on one aircraft, or he may transfer projects between aircraft to speed project completion. These variables are often interdependent, however, and may frequently produce conflicting measures of success against the various wing objectives. For example, while overtime will help project completion times, it will also cost more of the customer's money. Additionally, not all variables will affect all projects. Authorizing overtime in the modification center will not help meet any objectives if there are no aircraft available to modify. A scheduling decision, then, may become a tradeoff between the attainment of the several goals of the organization.

Wing Resources. To implement any improvements to the 4950th Test Wing scheduling process, one must assume the wing will not be able to increase resources except in the area of computer support. The resources currently available to the wing include nearly 2000 individuals, 45 aircraft, and office, hangar, and maintenance facilities sufficient to accomplish their flight test mission (Glenn, 1984:6-9). The wing currently owns two Digital Equipment Corporation VAX 11/750 computers designated for housing a new wing-wide information management system. The computers will be linked by an ethernet system and will allow access to several external data storage devices. The wing has already contracted for the Oracle database management system and the EIS graphics system for use with the management information system. Any computer aided solutions must initially operate within these systems (Test Wing, 1985:6).

## Statement of the Problem

The wing is developing an information system to update data regarding the current state of projects within the wing; however, a clear and accurate presentation of this data for decision making purposes has not been implemented. For the purposes of this research, data refers to a quantity of raw facts, while information refers to the meanings assigned to the data (Morris, 1985: 11; Rogers, 1985). Thus, for project data to be useful in the decision making process of the test wing, it must be presented to the commander in a form which provides insight into project impacts on wing resources, capabilities, and goals. Currently, the information requirements for project management, the data requirements implied, and the methods for processing the data and presenting the information in useable forms do not exist.

## Research Question

What data must be collected and maintained, how should it be processed, and how should the resulting information be presented to the commander for him to adequately assess the impact of a project on the wing schedule and resources?

## Subsidiary Questions

This study breaks the research question into several manageable sub-questions. The approach begins with the intended result and works backwards to determine items required to yield the intended result.

1. What are the goals and quantifiable objectives of the Wing with regard to project management?
2. What impacts can a project have on the wing schedule and resources?
3. How can the Wing control or vary these impacts?
4. What criteria does the decision maker use to compare the various decision options?
5. What information does a decision maker require to make project scheduling and resource allocation decisions?
6. In what forms can the information be presented to provide the decision maker an accurate and easily understandable picture of his schedule and allocation options?
7. What data is required and how can it be processed to yield the necessary information?

As a prelude to answering these questions as they relate specifically to the 4950th Test Wing, this study identifies the overall methodology selected as best suited for this type of decision making problem by assessing decision support opportunities of several project management and scheduling methodologies.

## II. Historical Review of Scheduling and Program Management Techniques

### Introduction

The obvious problem of the 4950th Test Wing addressed in this study is one of scheduling and project management. Underneath the surface, however, one finds the root of the problem to be in the generation and use of information for choosing between possible alternative schedules for the best accomplishment of wing goals. This chapter presents an overview of the history of project management and scheduling techniques from the management developments of Gantt charts and program review techniques (PERT and CPM), through mathematical and heuristic scheduling advances, to the incorporation of the above into systems focused on the generation and use of information specifically for decision making.

### The Beginnings of Project Management

In the Beginning. As early as the 19th century, a few men recognized the need for business management, as shown in the words of Charles Babbage in On the Economy of Machinery and Manufactures," in 1832:

> A manufacturer . . . must attend to other principles besides those mechanical ones on which the successful execution of his work depends; and he must carefully arrange the whole system of his factory in such a manner, that the article he sells to the public may be produced at as small a cost as possible. (Dale, 1965: 146).

However, development of business management theory and implementation of cost reduction techniques was not wide spread. Businesses were generally small, and owners could manage their affairs through common sense. The main skill required of a successful businessman was a knowledge of the manufacturing processes involved or the tasks to be performed on a job (Dale, 1965: 147).

Gantt Charts and Managing Work Flow. The first major attempt at managing the flow of work in a project was by Henry L. Gantt with the employment of a chart for tracking project progress. A Gantt chart is a horizontal bar chart plotting activities on the vertical axis against time on the horizontal axis (Figure 2.1). It provides a quick overview of the status of the organization and the progress of individual activities.

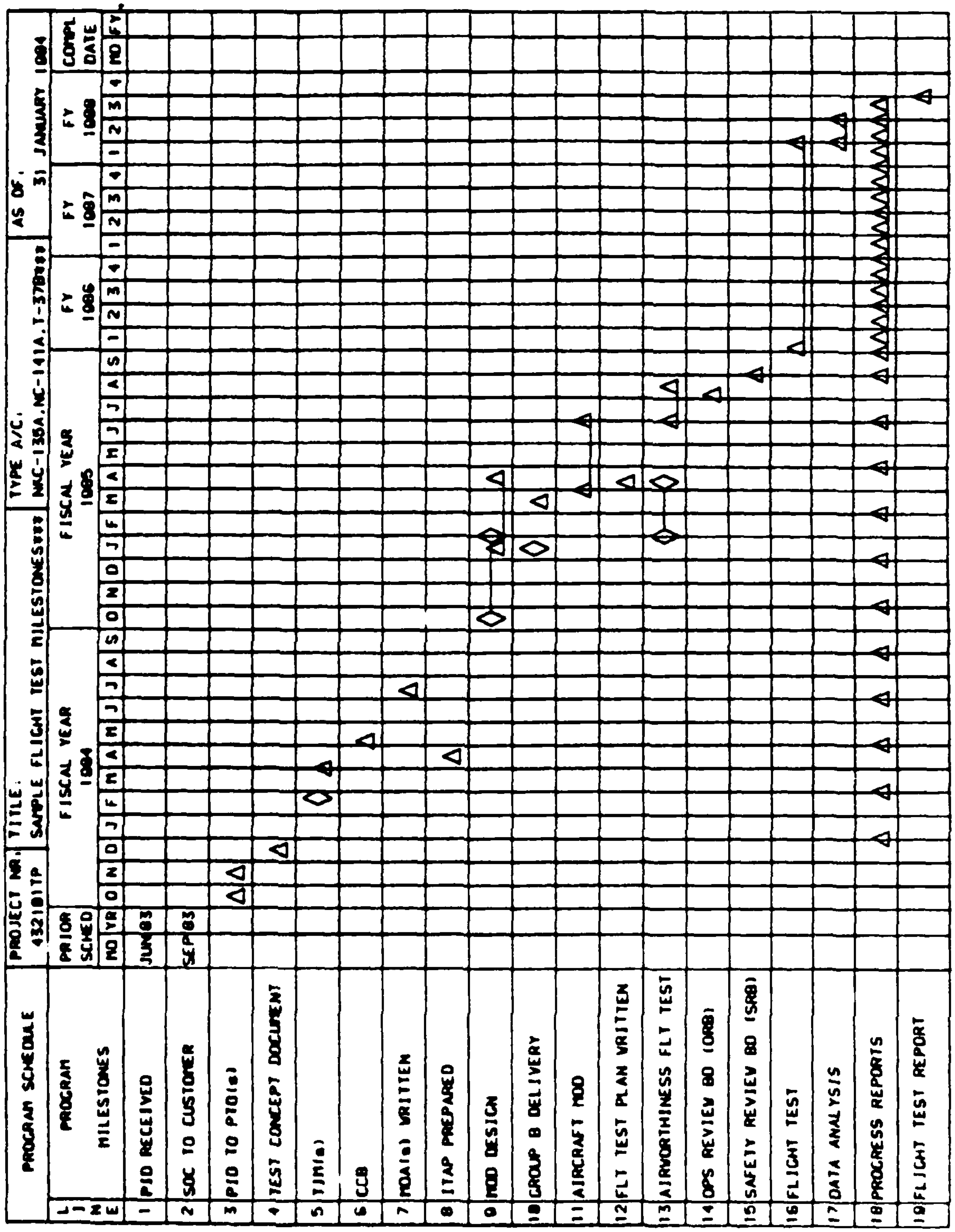



Figure 2.1 Sample Gantt Chart Representation of a Project Schedule (Interviews, 1985)

Uses of Gantt Charts. Gantt charts can be arranged to show more than just work progress. As shown in Figure 2.1, by adding symbols, the charts can track milestones and changes to original schedule estimates. Multiple lines can be used to represent resource and manpower utilization, allowing a supervisor to gauge requirements. Color can be added to aid in separating data by project, work area, or required skills and resources. The flexibility of Gantt charts has made the visual aids very popular for laying out and tracking project schedules (Gavett, 1968: 537).

Disadvantages of Gantt Charts. While Gantt charts can provide a quick view of project status, they have several disadvantages when used for making decisions about the schedules of large organizations like the 4950th Test Wing. Creation of alternative schedules can be difficult, requiring the physical movement of chart lines. Even if the movement can be automated through computer graphics, Gantt charts do not readily show interrelationships between activities: activities which must be completed prior to other activities (McGough, 1982: 76). Thus, the user might not recognize all of the effects of a schedule change. Additionally, Gantt charts do not readily allow indexing of information: simultaneously tracking resource and manpower utilization by shop or activity to avoid shortfalls, resource and manpower utilization by project to identify potential problem areas, and project progress through activities to track the accuracy of the schedule. While deviations from the schedule may be easily spotted, without knowledge of activity interrelationships the user cannot readily identify future effects.

PERT and CPM. The Program Evaluation and Review Technique (PERT) was developed in the 1950's to aid in managing activities and tracking progress on the massive Polaris Submarine project. The basis of the technique is a network depicting all activities required to complete a project and the interrelationships between activities. Figure 2.2 shows a simple example of how PERT might be applied to a project at the 4950th Test Wing. Beyond tracking project progress as a fancy Gantt chart, the main use of PERT is in determining the probability of completing activities and projects on schedule. While the mechanics of those calculations are left to texts devoted to the subject (for

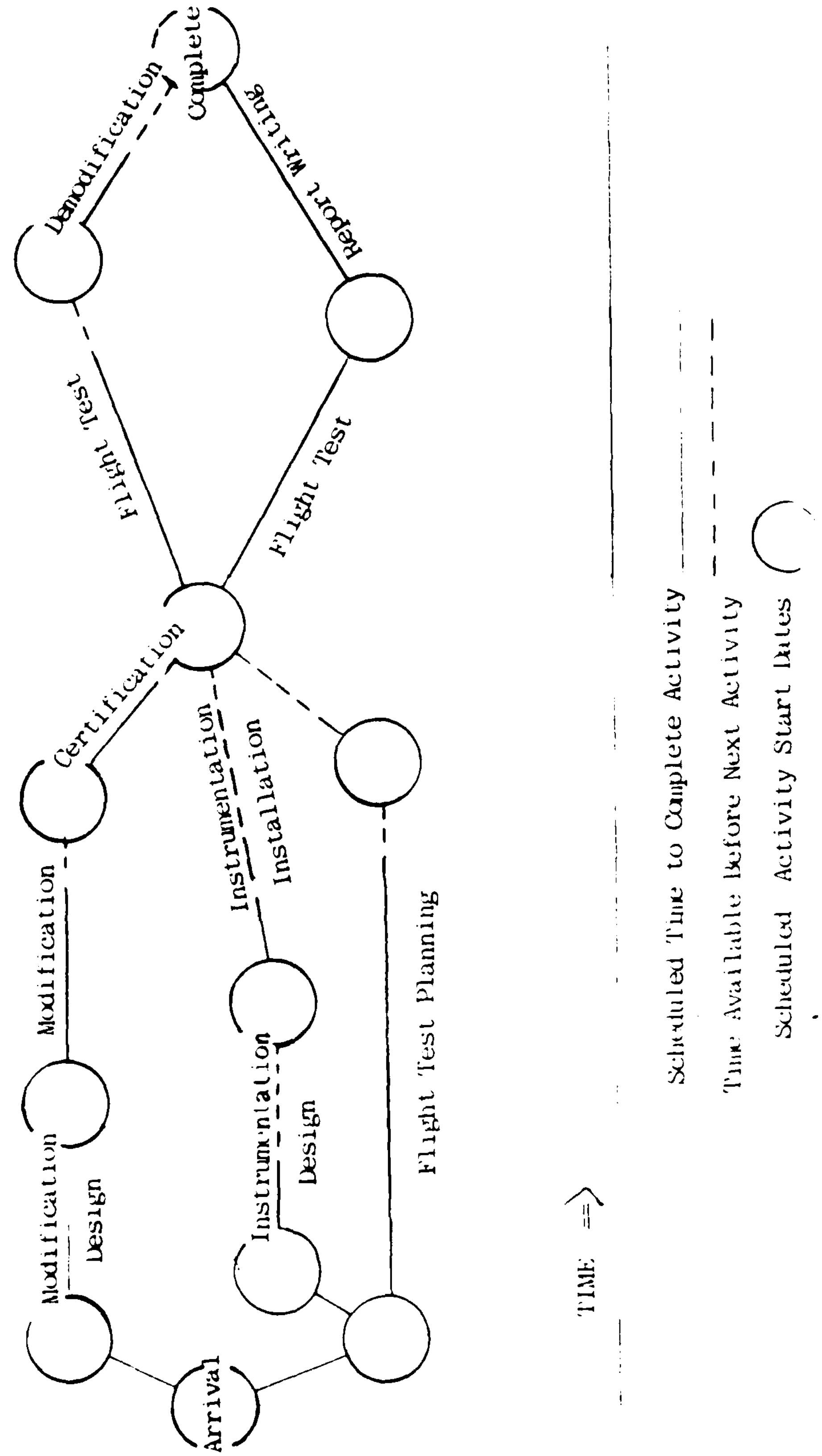



Figure 2.2 Sample PERT/CPM Representation of Project Flow (Interviews, 1985)

example, Hillier, 1980: 246-259), suffice it to note that the technique can identify the expected start and stop dates for each activity, allowing planners to more efficiently schedule workers and resources.

The Critical Path Method. The Critical Path Method (CPM) is another network based project management tool. It is frequently used in conjunction with PERT to identify activities in which schedule deviations will affect overall project completion (the critical path). CPM also allows consideration of trade-offs between cost and time; for example, an activity may be finished quicker if employees work overtime, however the overtime pay will add to the cost of the activity. The question of which activities should be expedited in order to meet a given project deadline at minimum cost can be answered through mathematical programming. Again, the mechanics of the calculations is left to the many texts on the subject (for example, Hillier, 1980: 253-7). In the area of project management, however, CPM finally adds a goal (minimum cost) to scheduling and admits that managers may be able choose between schedules - trading cost for time.

Disadvantages of PERT/CPM. While PERT and CPM provide a good starting point for project management and scheduling by forcing the manager to structure the flow of work in the project and consider interrelationship between activities, there are at least three deficiencies in applying the techniques to the 4950th Test Wing. The first shortcoming is the implicit assumption of unlimited resources (Cooper, 1976: 186; Patterson, 1982: 1): if resources are not available, the organization may not be able to complete the project in the predicted time. The major cause of such resource limitations is competition between projects, which leads to the second shortfall: overlaying the schedule networks of the several projects underway in the test wing at any one time, and determining which projects must be delayed because of competition for resources, could result in a haphazard schedule since PERT and CPM do not directly consider any system of priorities between projects. The third deficiency lies in not considering organizational goals beyond cost and completion times.

Summary of Early Project Management Efforts. Early businessmen did not generally concern themselves with project management. The first attempts at business management were directed at inducements to labor and improving the work process. The Gantt chart introduced the concept of managing the flow of work in projects, however it lacked methods to show the effects of changes upon interrelated activities. PERT and CPM added the interrelationships between activities within projects, but still fell short in analyzing the interrelationships among projects. What is needed is a scheduling mechanism which takes into account limited resources and allows for competition between projects.

Schedule Generation Techniques

Introduction. J. William Gavett provides a textbook definition of scheduling as "specifying when, in calendar time, certain events are to take place (Gavett, 1968: 536)." When dealing with only one project, placing individual activities on a calendar might seem a trivial task, especially after introducing PERT and CPM techniques; however, in a large multi-project organization like the 4950th Test Wing, not all activities may be scheduled at the times identified by their individual CPM networks: one must now consider competition between activities of different projects for limited resources. Obviously, if two activities both require one special worker at the same time, one activity must be delayed. If both activities are on the critical paths of their projects, the delayed activity will result in an overall project delay. An effective scheduler, then, must somehow decide which activity and project to delay. Two general approaches have evolved which allow the scheduling decision to be put into the context of achieving the overall goals of the organization: operations research/management science (OR/MS) optimization techniques, and job shop heuristic-based techniques.

Optimization Techniques. The aim of traditional operations research/management science (OR/MS) techniques as applied to project scheduling is to find the schedule coming the closest to meeting some organizational goal. To accomplish this aim, OR/MS techniques attempt to reduce the problem into an exact mathematical form: a set of equations which can be solved mathematically in terms of quantifiable organizational goals such as the time to complete projects or the cost associated with delays. Goal programming is one such technique which has been applied successfully in work force planning and scheduling (Goodman, 1974; Lin, 1980; Zeleny, 1982: 300-6).

Goal Programming. In goal programming, one attempts mathematically to find the schedule that minimizes the deviations from quantifiable goals. Goal programming allows consideration of multiple, potentially conflicting goals. Its inherent limitations, however, frequently make goal programming unacceptably restrictive when dealing with real world problems (Lee, 1972).

Linearity. All objective functions, constraint equations, and goal relationships must be linear: twice the activity uses twice the resources. For example, if building 1 table takes 12 hours then building 2 must take 24 hours. No allowance is made for the ability to begin work on the second table while the glue dries on the first.

Divisibility. Goal programming also assumes all activities and resources are divisible. No worker would believe one man working for half of an hour accomplishes the same amount of work as half of a man working for a full hour.

Deterministic Quantities. The deterministic assumption implies all parameters are known. In the 4950th Test Wing, each project is unique: resources and work times can only be estimated.

Goal Deviations. With the objective of minimizing the sum of all deviations from stated goals, goal programming assumes these deviations can somehow be equated between goals: deviations from the goals can be presented in similar terms (hours, dollars, etc.) to allow their addition in the objective function. In the 4950th Test Wing, one finds no consistent relationship between overtime and lateness.

Other OR/MS Techniques. Some of the limitations to goal programming may be avoided by using other OR/MS techniques. Non-linear programming techniques can eliminate the problems associated with the linearity assumption. Integer programming techniques can reduce problems associated with divisibility. These techniques, however, reduce limitations only at the cost of greater model complexity, increased computational difficulties, and increased time to run the model and generate the desired schedule (Cooper, 1976: 1186). In general, large real world problems tend to have too many possible combinations and intricate complications for efficient mathematical programming (Davis, 1975: 944), rendering the exact methods exemplified by traditional OR/MS techniques unrealistic. Traditional OR/MS techniques require reduction of all processes into exact mathematical formulations and do not allow for qualitative inputs or judgements.

Heuristic-Based Job Shop Techniques. A heuristic, in the context of problem solving, is a rule of thumb: a reason or method that works, regardless of theoretical support. Job shop scheduling techniques use heuristics to set priorities - which activities will be worked on first in the event of competition for limited resources. A simple job shop scheduling algorithm begins at the top of the priority list and enters activities onto the schedule calendar so long as resources are available (Patterson, 1982: 4). The result is not necessarily the best schedule, but a good schedule balancing the rule of thumb against achievement of the organizational goals.

Research Toward Determining Good Rules of Thumb. Rules of thumb, or priority rules, determine the order in which jobs are worked. Early research in the job shop scheduling field focused on finding the priority rules which performed best against specific measures of performance. In the 1960's Richard Conway and his associates used computer simulation to test five common priority rules against varied measures of performance (Conway, 1960, 1960a, 1960b). Their tests found no single priority rule to maintain consistently good overall performance against varied measures of performance: each measure apparently had a corresponding rule for best performance (Conway, 1960a: 124). Conway's results have been variously confirmed and disputed in the ensuing two decades (see Patterson, 1982 and Davis, 1973 for reviews). Patterson maintains the most likely reason for such disagreement is the lack of a consistent set of data (Patterson, 1982: 4), which he attempts to solve in his 1982 monograph. A significant advance in this area was made by John Dumond, who used the Patterson data base to reevaluate several rules previously studied (Dumond, 1985).

Limitations of Job Shop Scheduling. The schedule provided by job shop scheduling techniques is based on the chosen priority rule, rather than on the exactness of OR/MS mathematical formulations; however, it is still only one schedule and provides the decision maker no choice. As Conway discovered, single priority rules do not perform equally well for different measures of performance; thus, in choosing only one priority rule, the job shop technique may bias the resultant schedule toward only one organizational goal. Additionally, strict job shop scheduling only schedules and does not incorporate facilities for project management: managing the work flow by identifying potential bottlenecks or periods of slack; testing the effects of changes in times, goals, or resources before changes are made; and identifying downstream effects after changes are made.

Computer Simulation in Job Shop Scheduling. Computer simluation appears at first to offer some relief to the restrictions of job shop scheduling algorithms; however, on closer examination, one finds that for the purposes of project management and scheduling as examined in this study, computer simulation still has limitations in its ability to directly aid decision makers. Computer simulation has two main uses in project management and scheduling: generation of statistical data and generation of random schedules. Simulation generally involves allowing a large number of projects, each with statistically determined shop times and resource requirements, to flow through a computer model of the organizaiton. Managers may observe the numbers of projects waiting for work at each shop, the time projects spend waiting to be worked, and the time spent waiting for other prerequisite activities to be finished. From this statistical information, the managers can locate potential bottlenecks: where too many projects are waiting for too long a time. Unfortunately, such statistical information does not help managers generate schedules for projects at hand. Simulation can, however, be used to generate schedules of a sort. Once an accurate model of the organizaiton is built, it can be used to predict the results (in terms of resource use and completion dates) of allowing a given set of projects to flow through the organizaiton in any particular pattern. Thus, by randomly varying the relative priorities of projects in each simulation run, the new flow patterns generated can result in hundreds of possible schedules. The problem for the manager is in selecting the "best" schedule for implementation. Computer simluation, as with more traditional scheduling techniques previously discussed, has no intrinsic ability to aid the manager in this most important decision.

Summary. Both OR/MS and job shop techniques provide single schedules. While useable, these schedules may be biased due to the limitations of mathematics in describing real world operations (OR/MS) or the choice of a single priority rule (job shop). Additionally, these single schedules assume that all inputs are known in advance and that no changes or delays may occur; they provide the decision maker no opportunity to test the effects of changes and error. While additional schedules using different inputs could be generated, OR/MS and job shop techniques, including computer simulation, provide no means to directly compare the additional schedules in terms of accomplishment of organizational goals or of the likelihood of delays. To combine both scheduling and project management, one must progress from generating THE answer to providing scheduling information to the decision maker.

## Information Systems for Project Management

Advances in Information Systems. With the arrival of the computer in business and industry, computerized information systems have evolved from electronic data processing (EDP), through management information systems (MIS), toward decision support systems (DSS). While the following descriptions of these information system types, taken mainly from Sprague and Carlson (1982), are by no means definitive with clear cut and easily recognizable boundaries, they do provide a useful framework for dicussing how information has been viewed and used in business.

Electronic Data Processing. The first form of computerized information use in business was electronic data processing (EDP). EDP centered on transaction processing, accounting, and generation of summary reports (Sprague, 1982: 6). While EDP made many routine daily business functions easier, in terms of information for decision making it provided little more than a periodic review of what transactions had been made in the past accounting period.

Management Information Systems. Management information systems (MIS) try to integrate the flow of information in an organization (Sprague, 1982: 7). Generally, MIS focus on a large database incorporating all the information the organization produces and uses in its operations. A classic example of MIS development is the Business Systems Planning (BSP) study accomplished by the 4950th Test Wing in 1984. The initial concept of the BSP is to recognize information as a resource which must be managed and made available throughout the organization (IBM, 1981: 1). The wing followed the BSP methodology and identified the creators, users, and flow of internal information (Glenn, 1984). The result was a massive database designed to allow easier access (entry and inquiry) to internal information and more efficient generation of routine reports. Because of the integration of the entire organization into the database structure, more information is available more readily to a decision maker; however, the focus is still on data and report generation, not on presenting the data in a form useable for decision making.

Decision Support Systems. The focus of Decision Support Systems (DSS) is on the decision maker (Sprague, 1982: 7). The evolution from MIS to DSS involves the generation and presentation of information in a form useable for decision making, that is for making trade-offs between organizational goals in choosing among alternative courses of action.

Summary of the Information Revolution Evolution. The differences and contributions of EDP, MIS, and DSS toward the effective use of information can easily be lost in semantics. As defined here, their comparisons can best be summed up visually, as in Figure 2.3.

Decision Support System Integration of Concepts. A DSS integrates many of the features of the previous dicussed project management and scheduling techniques. It incorporates the database concepts from MIS, the use of models from OR/MS and job shop techniques, and the use of graphic representations of information from the early days of project management. In addition, DSS incorporate the idea of interaction with the user to allow the decision maker to control the decision making process: the generation and comparison of alternatives leading to a final decision.

| | EDP | MIS | DSS |
|---|---|---|---|
| USER | Accountant | Middle Manager | Decision Maker Analyst |
| TASK | Structured: Accounting Administration | Structured: Operations Control | Unstructured: Analysis Planning |
| Development Approach | Data Files | Total System | Adaptive |
| Emphasis | Data (Product) | Integrated Data | Decision Process |
| Contribution | Start | Integration Data Base Mgt Data Dictionary | Relational data bases Model Mgt |

Figure 2.3 The Information Revolution Evolution (Valusek, 1985)

<u>Examples of Concept Integration</u>. An example of DSS integration of techniques is shown in the case of the Southern Railway Company. The company instituted a DSS to aid the track superintendent in making train passage decisions: determining which trains to hold at which sidings when two or more trains meet. Their database included the current status of the railroad and of all operating trains. The basic model used a branch and bound algorithm to determine best routings. The graphics displays included four television screens: two for displaying the track layout, one as a worksheet for updating the train data files, and one for testing various routings. The interaction capability allowed the superintendent to respond as needed to changing conditions such as train speeds, and track closures. Further, the system allowed the superintendent to ask "what if" questions to examine the overall effects of decisions before they were implemented. In sum, the DSS used the current status of the railroad to generate a routing, then allowed the superintendent to generate additional routings based on his professional judgement and experience, and finally allowed the superintendent to examine the effects of all the routings before making a final decision. The DSS reduced superintendent workload in making train passage decisions, and has resulted in an overall decrease in train delays throughout the system - a major goal of the company (Sauder, 1983).

DSS Integration in the 4950th Test Wing. A DSS could integrate project management and scheduling in the 4950th Test Wing. The planned MIS database is to include the current status of all projects in the wing, as well as the status of wing resources. The model base could use a relatively simple job shop scheduling algorithm to generate overall wing schedules. The interaction capability would allow the decision maker to change resources, project requirements, and other internal data in generating additional schedules based on experience, judgement, and "what ifs" regarding likely delays or changes. The graphics presentations could then display not only the individual schedules, but also the effects of the various schedules as they apply to important wing goals. A DSS should help solve the wing information, project management, and scheduling problems.

## Summary

Early project management techniques were unable to incorporate simple methods for scheduling multiple projects when faced with limited resources. Job shop and OR/MS scheduling techniques by themselves were unable to incorporate important project management functions. Additionally, neither early project management or scheduling techniques approached the problem from an informational and decision making point of view. Early information systems techniques tended to focus on the accumulation of data and generation of routine reports. Decision Support Systems combine many of the useful aspects of these earlier project management, scheduling, and information systems techniques, along with the concept of user interaction to allow for generation of and choice between additional alternatives based on non-quantifiable factors. DSS may offer a useful solution to many of the 4950th Test Wing project management and scheduling problems. Because DSS are a relatively new concept, Chapter III will discuss more fully the what, how, and why of DSS, along with a general plan for their design.

## III. Decision Support Systems

### Introduction

To fully develop the subject of Decision Support Systems (DSS) is an ambitious undertaking; entire books have been written about it. At the same time, it is impossible to establish the requirements for a particular DSS without first establishing the basic concepts that are it's foundation. This chapter addresses some of the key concepts surrounding DSS and lays the groundworks for the specific DSS that will be described in chapter IV. First, DSS will be defined in terms of what they are, what they do and where they can best be employed. Then, the decision making process will be addressed with respect to the components of DSS and how they support this process. Finally, an approach to designing and building DSS will be presented and described.

### Defining Decision Support Systems

Definition. In the broadest context, a DSS can be thought of as a mechanism that provides information to help a manager make a choice. Key to this description is the word "information." As opposed to data (which includes raw facts, tables of numbers, lists of names, dates and places, etc.), information is the meaning attached to facts, numbers or lists (Morris, 1985: 11; Rogers, 1985). The idea is that data becomes information when it is related to a situation or problem and is presented in a form that provides meaningful insight into making an assessment or a decision.

Information. In the context of DSS, information is a meaningful display of data (generally in tables or graphs) depicting how well alternatives achieve underlying goals in light of changes in the operative variables. The value of a DSS, then, is that it provides the means by which a manager can obtain information, such as possible results of alternative actions, view this information in a form that relates it to the underlying goals, and make decisions based upon objectives that he is trying to accomplish.

Problem Structures. In addition to information, the concepts of "structured," "semistructured" and "unstructured" problems are important in further describing situations in which decision support systems are especially useful. In a structured problem, specific series of formulas or decision rules can be employed to identify that a problem exists, to develop possible solutions, and to choose among the alternatives. Consequently, structured decisions often do not require the attention of a manager since the decision process is understood to the point that it can be relegated to clerical help or computer automation. In contrast, the solution process for unstructured problems cannot be (or has not been) fully defined and thus requires the judgement of a manager. The middle ground of semistructured problems includes those where some of the problem identification or solution steps can be clearly delineated and relegated while others require the decision making judgement of a manager. (Keen, 1978: 86-95) Examples of the different types of problems are listed in Table 3.1.

TABLE 3.1

Comparison of Problem Structures (Keen, 1978: 87)

| | Management Activity | | | |
|---|---|---|---|---|
| Type of Decision/ Task | Operational Control | Management Control | Strategic Planning | Support Needed |
| Structured | 1 Inventory reordering | 4 Linear programming for manufacturing | 7 Plant location | Clerical, EDP or MS models |
| Semistructured | 2 Bond trading | 5 Setting market budgets for consumer products | 8 Capital acquisition analysis | DSS |
| Unstructured | 3 Selecting a cover for Time magazine | 6 Hiring managers | 9 R & D portfolio development | Human intuition |

When a decision process can be fully structured, the traditional techniques of electronic data processing (EDP), management information systems (MIS) and operations research/management science (OR/MS) can be applied to produce solutions to the specific questions at hand (Keen, 1978: 11). These techniques require that the problem be clearly definable and that the decision processes lend themselves to automation. In contrast, one of the key aspects of decision support systems is the focus on unstructured or semistructured decision environments. "Most, if not all of managers' key decisions tend to be fuzzy problems, not well understood by them or the organization, and their personal judgement is essential" (Keen, 1978: 58). Sprague and Carlson address the DSS operating scenario as follows: "A DSS should provide support for decision making, but with emphasis on semistructured and unstructured decisions. These are the types of decisions that have had little or no support from EDP, MIS, or management science/operations research (MS/OR) in the past" (Sprague, 1982: 26). Much of the associated literature contends that it is this type of semistructured or unstructured environment where decision support systems offer the greatest benefit.

DSS do not try to replace the manager through automated solution finding techniques; rather, their purpose is to support and enhance his or her decision making ability (Keen, 1978: 58). As discussed by Herbert Simon,

> Uncertainty, computational complexity, and lack of operationality have been the principle barriers to extending operations research techniques to the upper levels of management. Qualitative concerns often elude the classical OR models, since human thinking and decision-making do not depend on the presence of numbers in the way that OR techniques do (Simon, 1982: 36).

Traditional techniques of OR/MS are primarily aimed at producing optimal solutions in well defined scenarios. Decision support systems, however, provide a coherent strategy for going beyond these traditional problem solution techniques by allowing managers to inject qualitative judgement into the decision process (Keen, 1978: 11).

Definition Summary. Definitions of decision support systems range from the broad view of any system supporting a manager's ability to make decisions (Sprague, 1982: 4, Keen, 1978: 58) to the more restricted perspective that DSS are "interactive computer-based systems that help decision makers utilize data and models to solve unstructured problems" (Sprague, 1982: 4). Regardless of the terminology used to define DSS, the primary emphasis is on the concept of assisting the decision maker. DSS support, rather than attempt to replace, the manager (Keen, 1978: 58). They rely on the premise that managers are generally competent when provided adequate information in usable form. A decision support system, then, is a system (input, process, output), either manual or automatic, that supports the cognitive processes of judgement and choice (Valusek, 1985).

Having identified what a decision support system is, the next step is to address the process of decision making and describe how the components of DSS support this process.

## The Decision Process and DSS Components

The Process of Decision Making. Herbert Simon presented an interesting view of problem solving when he wrote, "If we possess all the relevant information, if we can start out from a given system of preferences, and if we command complete knowledge of available means, the problem which remains is purely one of logic" (Simon, 1982: 41). Unfortunately, the decision process is seldom so clearly defined. More often it is an iterative process of investigations and assessments. Simon described the process in terms of three specific steps (Simon, 1960: 2):

> Intelligence: Searching the environment for conditions calling for decisions. Raw data are obtained, processed, and examined for clues that may identify problems.
>
> Design: Inventing, developing, and analyzing possible courses of action. This involves processes to understand the problem, to generate solutions, and to test solutions for feasibility.
>
> Choice: Selecting a particular course of action from those available. A choice is made and implemented.

The full spectrum of decision support involves helping the decision maker in all phases of the decision process: investigating and identifying the problem, generating alternative courses of action, and selecting a plan of action from the alternatives (Young, 1983: 28).

DSS Components. To effectively assist the manager in these decision making steps, a DSS must possess three essential components: a data base of information relating to the decision scenario, a model base of available tools capable of manipulating the data to produce meaningful results, and a system of dialog that enables the user to direct the problem solving effort in terms of selecting applicable models to perform needed operations and then presenting the results of these operations in a sequence the user can relate to. Sprague and Carlson put it this way: "Dialog is the user-interface component. Data base is the memory component. Modeling is the analytic component. Integrating the three form a DSS" (Sprague, 1982: 301). The process, then, is that of a manager using data that has been processed by one or more models and displayed through the DSS dialog to identify problems, elicit alternative solutions and choose among them. The DSS, then, enables the manager to gather information (intelligence), iteratively investigate options and generate viable alternatives (design), and judge between the alternatives based on goals and objectives (choice).

The importance of the dialog component warrants emphasis. It is through the effectiveness of the man-machine interface that much of the success of DSS will be derived. From the user's vantage point, "the Dialog is the System. All the capabilities of the system must be articulated and implemented through the Dialog" (Sprague, 1982: 29). DSS may possess comprehensive data bases and incorporate sophisticated manipulation techniques; but, if they do not convey these capabilities in a form usable to the manager, or if they do not present results in meaningful manner allowing the decision maker's mental process to proceed without disruption, the potential value of the DSS is diminished.

## Designing and Building DSS

Iterative Design Process. Sprague and Carlson propose an approach to DSS design that recommends a modest initial effort and emphasizes continual evaluation and modification of the DSS (Sprague, 1982: 15,140). The first step is to select a workable subproblem. This "kernel" should be small enough that the nature of the problem as well as the decision support requirements can be clearly identified and yet should be important enough to warrant the effort to solve.

Once this initial problem selection has been made, a simple support system is designed and built to assist the manager in dealing with the problem. This first attempt gives the decision maker something to work with and react to. It provides a basis for judgements regarding future renditions of the system.

Having experimented with and used the initial system, the manager is in a position to provide feedback on the DSS in terms of its capabilities and usability. This is a crucial step since changes, deletions and expansions to the current system will be based on these evaluations. In the framework provided by Sprague and Carlson, the system should be evaluated based on the impacts of using the DSS: Does its use result in sound, timely and cost effective decisions? Does it assist in the decision making process? Do the users feel it is understandable, usable and accurate? Are the characteristics of the system (cost, responsiveness, availability, etc.) acceptable?

Based on the results of the evaluation process, changes can be made to the DSS in terms of replacements, modifications, additions and deletions that will better equip the system to suit the needs of the decision maker. Hereafter, the evaluating and updating processes are repeated until the system reaches the desired performance level.

Building DSS. Since the user's perception is a major ingredient in determining the success or failure of a DSS, it seems appropriate to approach DSS construction from the user's perspective. With their ROMC (Representations, Operations, Memory aids, and Control mechanisms) approach, Sprague and Carlson provide such an avenue (Sprague, 1982: 96). Their methodology focusses first on the output information, both content and form, that the decision maker needs in order to effectively address the problem. Thus, they keep the manager and his perception of the problem at the forefront of the DSS design process.

All facets of the ROMC approach support one primary objective: to provide the decision maker the information he requires to deal with the situation at hand. This emphasis on the manager, with conscious effort to avoid structuring or confining his or her decision making process, is the crucial characteristic of the ROMC methodology.

The justification Sprague and Carlson use for the ROMC technique is centered in their analysis of decision makers and, as such, contains much of the rationale underlying the concept of decision support as a unique approach to problem solving. Their findings are summarized as follows (Sprague, 1982: 98-99):

1. Managers have difficulty describing the process by which they arrive at a decision, however, they often rely on conceptualizations (pictures, charts, graphs, reports, etc) to make or explain their decisions.

2. Although the decision making process may be hard to describe, all activities in decision making can be classified into one of the three steps in the decision process (information gathering, alternative generation, or alternative selection).

3. A requirement of almost all decision makers is the need for memory aids (reports, hand written notes, mental memory joggers, etc.).

4. Even in similar decision making environments, the styles, skills, and knowledge of managers can vary widely.

5. Regardless of the nature of decision support they receive, decision makers expect to exercise direct, personal control over that support.

These findings are central to the decision support philosophy espousing the "descriptive" process of <u>how</u> decisions evolve over the "prescriptive" ideology that assumes there is a right way to make decisions (Keen, 1978: 22). They also provide the basis for Sprague and Carlson's ROMC approach to building DSS.

<u>Representations</u>. As stated earlier, the ROMC approach starts with the output that the decision support system should produce to support the decision process. Since managers rely on conceptualizations to make or explain their decisions, a support system should enable the manager to view relational concepts in fashions suited for the information being presented. These representations may take the form of aggregations (tables, graphs, charts, plots, maps) and may support any of the decision process steps of information gathering, alternative generation, and alternative selection.

DeSanctis suggests that there is no convincing evidence identifying one form of presentation to be superior to others and that the best data display method is probably dependent on the task to be accomplished by the user. The end result is that when relationships applicable to the decision scenario are identified and provided to the manager in useable form, then comprehension of the problem and decision quality should improve (DeSanctis, 1984: 468).

In addition to the system providing information to the user, another important process that can be accomplished through representations is the user providing direction to the system. This can be in the form of menus, question-and-answer sequences, command language instructions, input-output forms, or any combination thereof (Sprague, 1982: 199-205). Again, the actual format chosen should reflect the needs of the user and the task at hand.

Operations. As stated earlier, all activities (or operations) in the decision making process can be classified into one of the three steps of information gathering, alternative generation, or alternative selection. Operations, in the ROMC context, encompass the various means of processing decision related data into meaningful results. They are the tools available to the manager by which he can manipulate information into useful ingredients in the decision process.

Operations can include such activities as information gathering, data manipulation, statistical analysis, system optimization, alternative generation, alternative comparison, and so on (Sprague, 1982: 104,260). Any packaged capability to process decision related information supports "operations" in the ROMC approach to building DSS.

Memory Aids. As the name implies, memory aids give the decision maker the ability to recall information. In everyday practice, these can include scratch pad notes, office reports, staff reminders, memos, or anything that can serve as a reminder. In DSS context, memory aids usually take advantage of computer capabilities and include various means to store and retrieve information and to prompt the user to perform necessary actions. Sprague and Carlson list the following as examples of memory aids (Sprague, 1982: 104):

data base: from sources that are both internal and external to the organization.

Views: aggregations and subsets of the data base.

Workspaces: for displaying representations and preserving intermediate results as they are produced by the operations.

Libraries: for saving workspace contents for later use.

Links: for recalling information from one workspace or library for use in another.

Triggers: for reminding managers that certain operations may need to be performed.

Profiles: for storing default values.

These are all examples of memory aids that might be built into a DSS. They support the requirement of managers to have memory support mechanisms that keep previously derived, decision related information readily available for use in the decision process.

Control Mechanisms. In the words of Sprague and Carlson, "The DSS control mechanisms are intended to help decision makers use representations, operations, and memories to synthesize a decision-making process based on their individual styles, skills, and knowledge (Sprague, 1982: 106). Control mechanisms provide the direct link between the user and the decision support system. They provide the means by which the manager actually directs the problem solving effort and therefore can be the critical determinant in how "user friendly" the system is perceived to be.

Control mechanisms can be of several forms. They can facilitate the actual use of the DSS such as function keys, command language instructions, "help" commands, and error messages. They can assist combining of several DSS activities into single joint activities or enable the user to alter representations such as adjusting graph scales or relabeling axes (Sprague, 1982: 106-107). In short, control mechanisms enable the manager to use the entire decision support system.

Thus, Sprague and Carlson's ROMC approach is a user oriented method of developing decision support systems. It requires the builder to look at the requirements of the user, throughout all phases of the decision process, and from this to determine the capabilities that must be incorporated into the DSS.

## Consolidated View of DSS Concepts

The decision process, the components of decision support systems, and the approach to designing DSS are three of the the central DSS themes presented in this chapter. The decision making process includes gathering information, generating viable alternatives and selecting among them based on goals and objectives. The DSS components (dialog, model base, data base) provide managers with the tools necessary to successfully negotiate the decision process in addressing a specific problem. The ROMC approach to DSS design provides a framework for identifying the system requirements (representations, operations, memory aids, control mechanisms) that enable the full range of decision support across all three phases of the decision process and within the capabilities of the three DSS components. The interrelations of these concepts are illustrated in Figure 3.1.

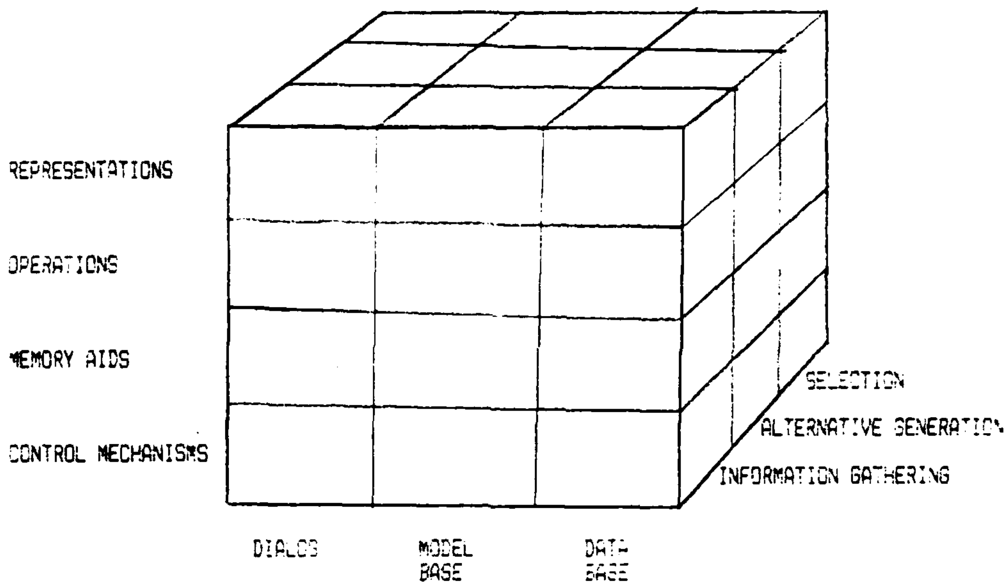



Figure 3.1 Interrelations of the Decision Process, DSS Components and ROMC (Valusek, 1985)

Although each of the individual concepts of decision process, DSS components and ROMC are valuable in themselves, it is the interrelationships between the concepts that are most valuable in DSS design. By analyzing each intersection of the three concepts (indicated by each block of the three dimensional cube of Figure 4.1), a DSS designer can be assured of addressing all facets of the specific decision support system at hand.

In reality, not every block requires individual attention; rather, only those intersections that have logical bearing on the problem scenario need be addressed separately. To illustrate, it is useful to view Figure 3.1 from two DSS perspectives: user and builder. From a user perspective, only the visible DSS component (dialog) is consequential while the model base and data base components have very little observable value. The user is concerned with how the dialog (through representations, operations, memory aids and control mechanisms) supports the decision process phases of information gathering, alternative generation and selection. In contrast, from the DSS builder's perspective, the ROMC relationships with the three components of dialog, model base and data base are of paramount importance while the underlying decision process is of little direct concern since it is primarily a function of the user. While neither is incorrect, it is the union of both perspectives (user and builder) that determines which blocks of Figure 3.1 require extensive consideration.

Perspective

The intent of this chapter has been to present some of the key concepts surrounding decision support systems and to provide a framework for building an effective DSS. Clearly, the emphasis is on the decision maker and the specific support ingredients that can help him or her deal with the decision scenario at hand. The next chapter attempts to integrate and apply these thoughts and methods into a conceptual system that deals with the specific problem environment of the 4950th Test Wing.

# IV. A Specific DSS for the 4950th Test Wing

## Introduction

The problem facing the 4950th Test Wing is: How can managers adequately assess the impacts of a project on the wing schedule, investigate options to create reasonable alternatives, and decide upon an effective course of action? For any resultant course of action to be viable, it must be consistent with the wing goals. As presented in Chapter I, the wing goals include maximizing the number of projects completed as well as the quality of testing provided while minimizing overtime requirements and due date delays. To achieve these goals, test wing managers can control only a limited number of operative variables: work capacity, project schedules, modification procedures, aircraft utilization, relative priorities among projects and the extent of testing accomplished.

To be an effective tool for decision making, a complete decision support system for the test wing must address each variable to determine its impact on any given situation. In the spirit of iterative design, however, this complete DSS is the end product, the ultimate aim of several iterations in the DSS development. The immediate requirement is to select a smaller "kernel" problem to address.

The specific purpose of this research effort is to present the requirements for a kernel DSS to deal with the manhour issue: how to best incorporate a new project into the wing schedule or adapt to changes in an existing project to minimize overtime and, at the same time, keep the work force gainfully employed.

This chapter will specify the decision support system requirements necessary to address the manhour issue. The chapter organization will follow the relational principles of the cube (ROMC approach, DSS components, decision process) presented in the previous chapter [see Figure 3.1]. Specifically, the dialog component will be analyzed from the user's perspective and in terms of the representations, operations, memory aids and control mechanisms needed to support the decision process of information gathering, alternative generation and selection among alternatives. Then, the model base and data base components will be addressed in terms of ROMC to identify the capabilities required to support the dialog component.

## DIALOG

The dialog component of a DSS is the interface between the user and the computer equipment with its software capabilities. It comprises the user inputs (menu selections and keyboard inputs) that direct the DSS to perform needed operations as well as the output (graphic displays) that the manager will use as a basis for his decisions. To establish the dialog requirements, each element of ROMC will be discussed.

Representations Applied to the Dialog. Representations include the graphical relationships that enable managers to acquire information about possible problem areas, to devise viable alternatives and to choose among them. The following relationships are important in addressing the manhour issue:

1. Project schedules.
2. Comparison between shops of forecast manhour utilization.
3. Manhour commitments versus capacity for a particular shop.
4. Impacts of a project on the manhour resources of a shop.
5. Projects competing for the same shop manhour resources.
6. Comparison between projects of manhour commitments for a shop.
7. Operative variables (things that might be changed) for projects competing for the same shop manhour resources.
8. Results of changes in project operative variables in terms of manhour commitments.

While these relationships overlap, they can be divided into three categories supporting the phases of the decision process. In general, relations 1 through 6 support the information gathering phase, 7 aids in the generation of alternatives, and 8 provides the comparison of alternatives enabling the manager to choose among them.

Information Gathering Representations. Test wing managers addressing the manhour issue may need to investigate relationships between projects (with their schedules and associated manhour requirements) and shops (with their work force capacity limits). Figures 4.1 through 4.6 are examples of representations that support this information gathering phase.

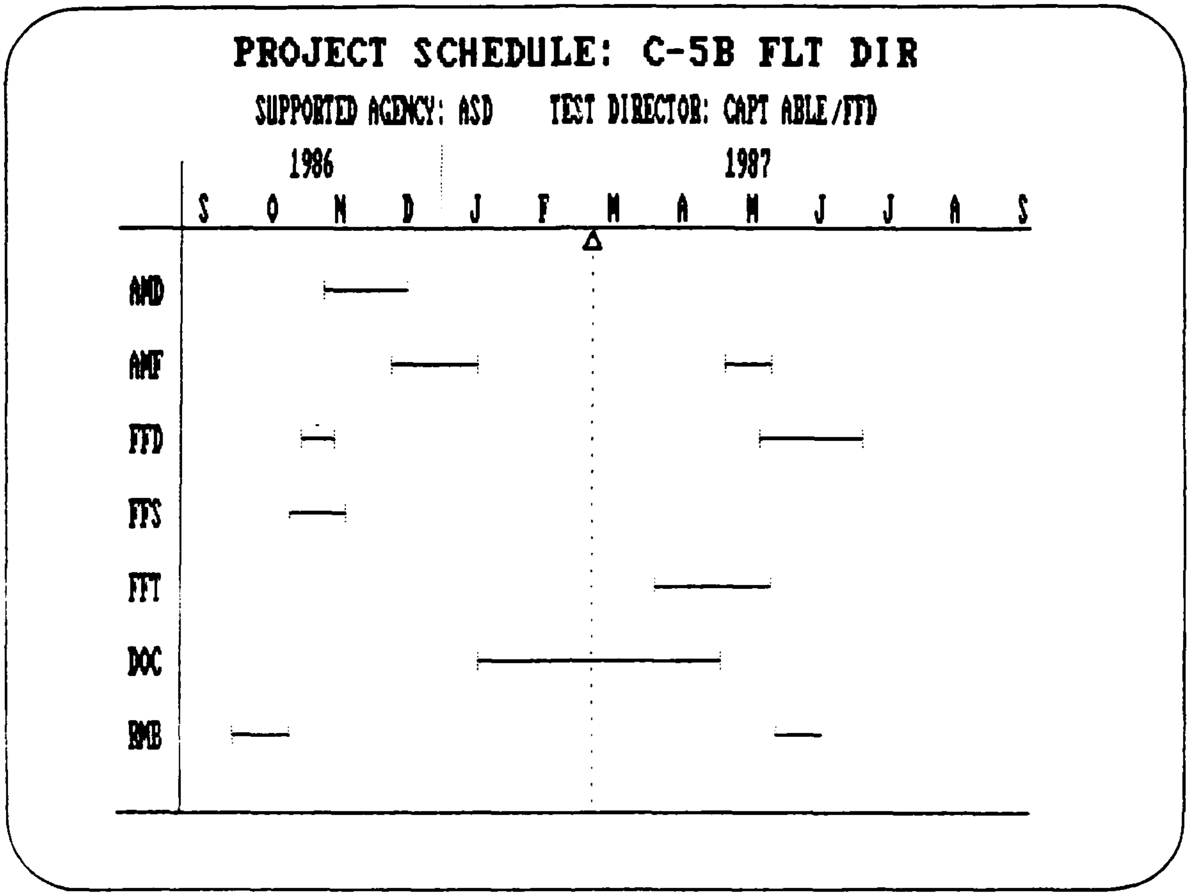



Figure 4.1 Project Schedule

Figure 4.1 shows a Gantt chart schedule for a particular project as it proceeds through its testing cycle. It shows the shops involved with that particular project and the flow of activities required.

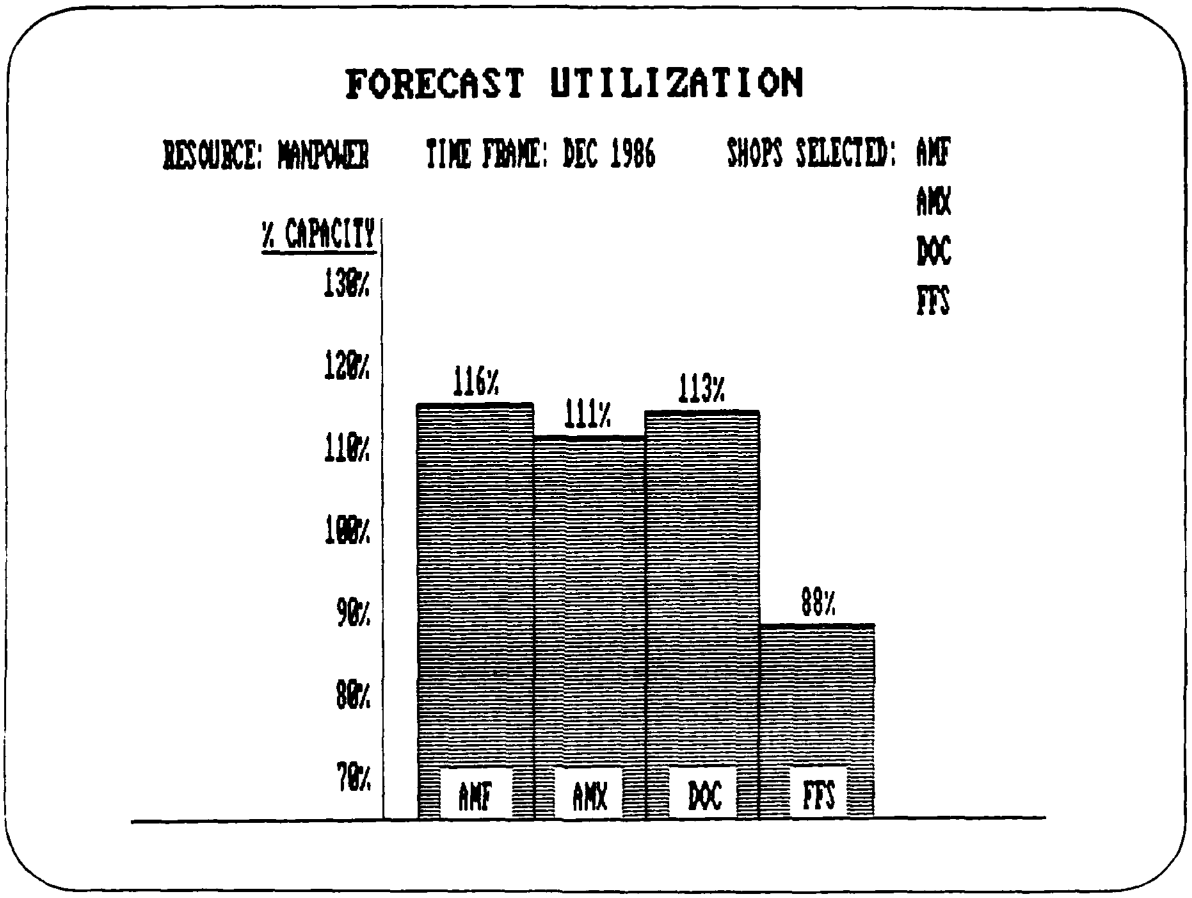



Figure 4.2 Manhour Commitment Comparison

Figure 4.2 shows a comparison between selected shops of forecast manhour utilization levels. It provides a graphic depiction of how heavily shops are committed in terms of their manhour capacities.

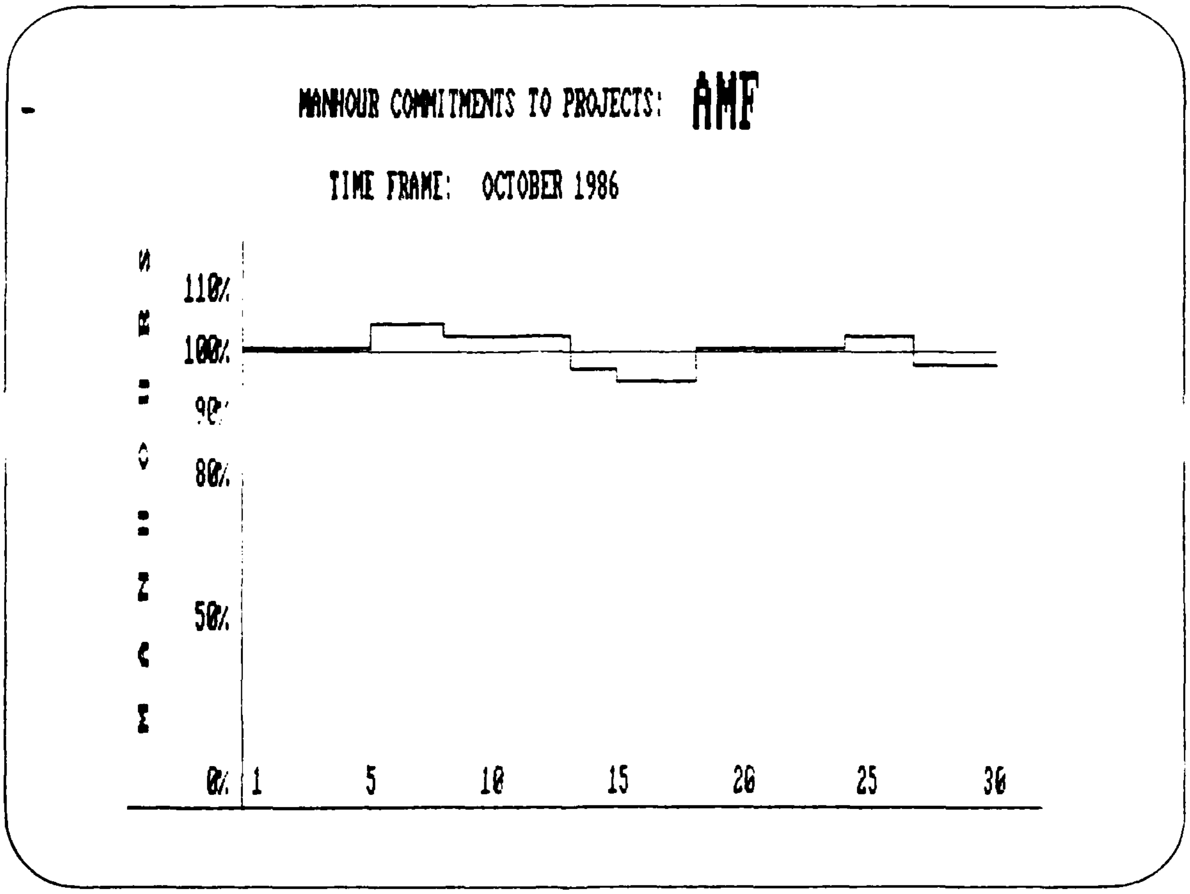



Figure 4.3 Manhour Commitments to Projects

Figure 4.3 relates total manhours committed for all scheduled activities of a shop to its manhour capacity over a period of time.

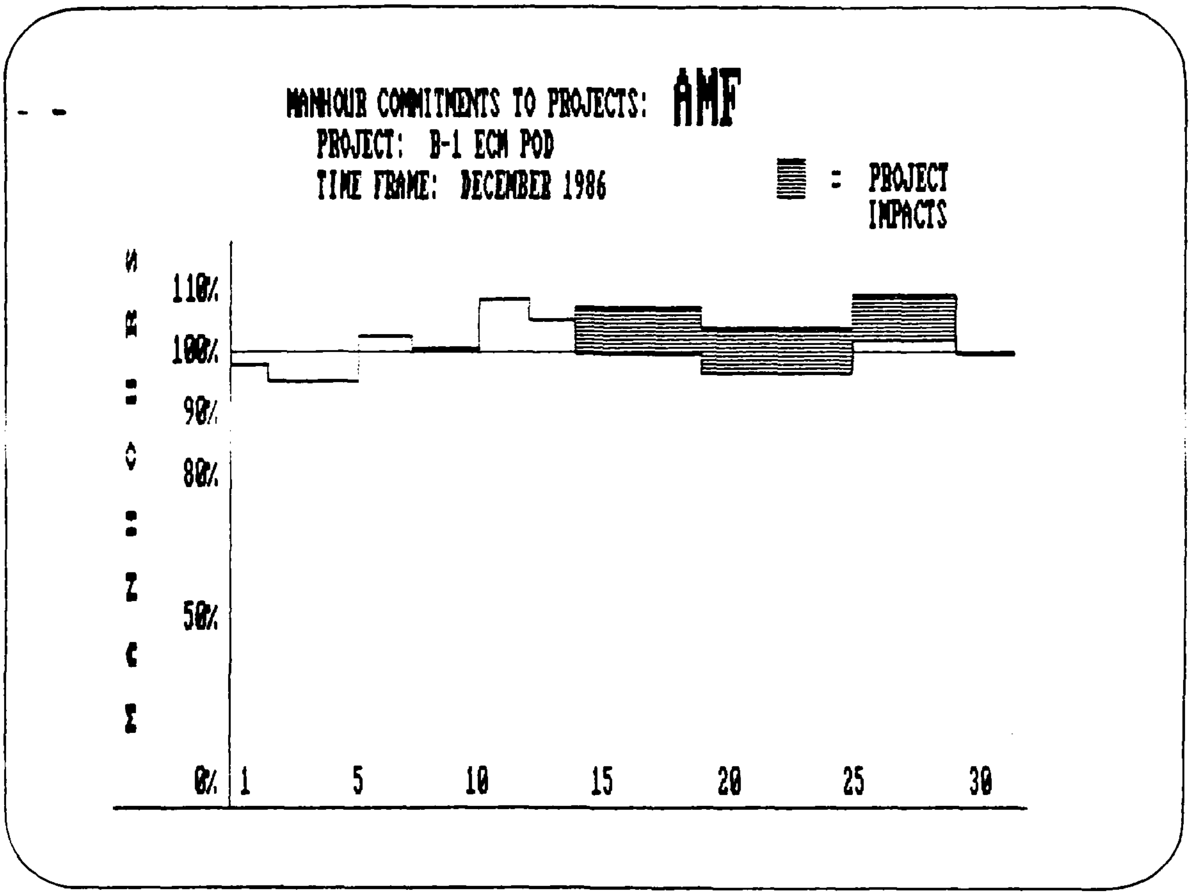



Figure 4.4 Project Impact on Manhour Commitments

In addition to providing commitment versus capacity information, Figure 4.4 gives a pictorial view of the impacts a particular project will have on the total manhours available for that shop.

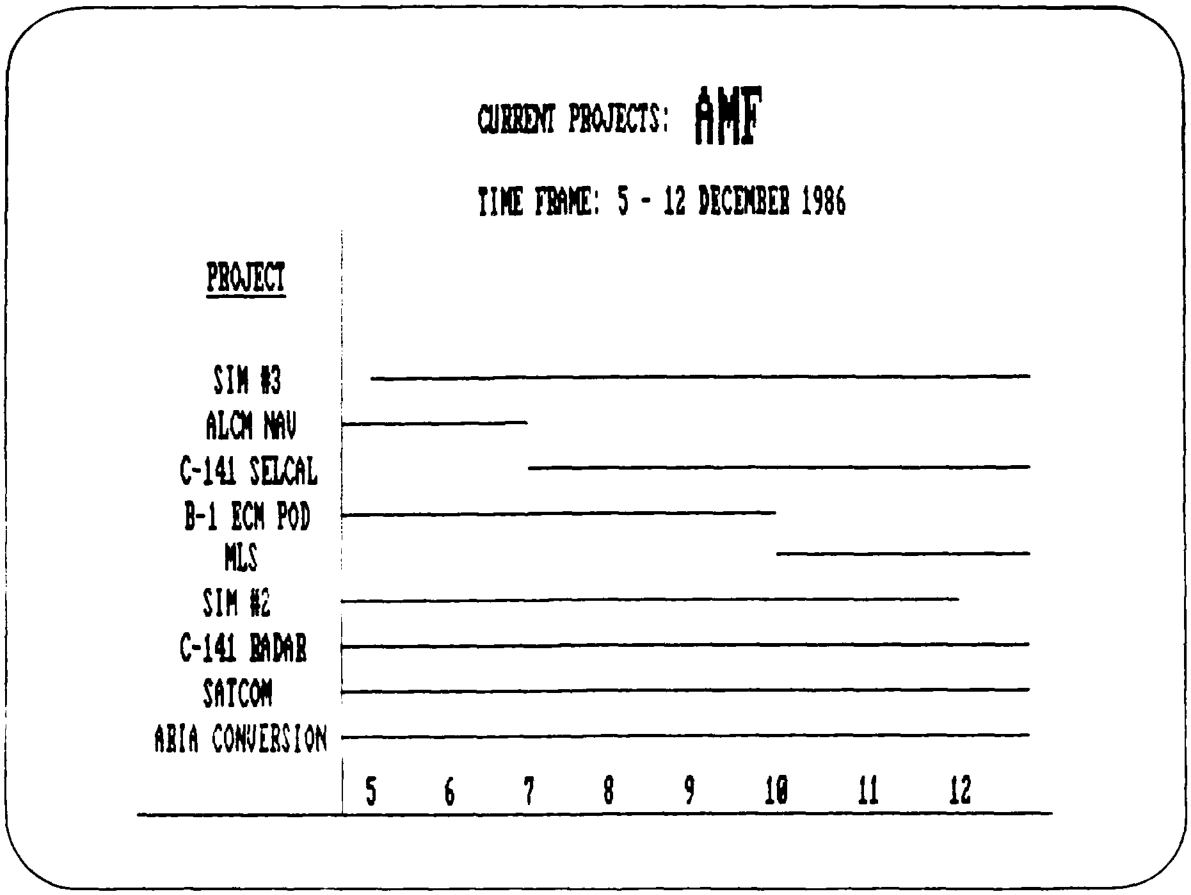



Figure 4.5 Projects Competing for the Same Resources

Figures 4.5 and 4.6 provide information regarding projects competing for the same shop resources. For a specified time frame of interest, Figure 4.5 shows the overlapping schedules of projects being worked by a shop.

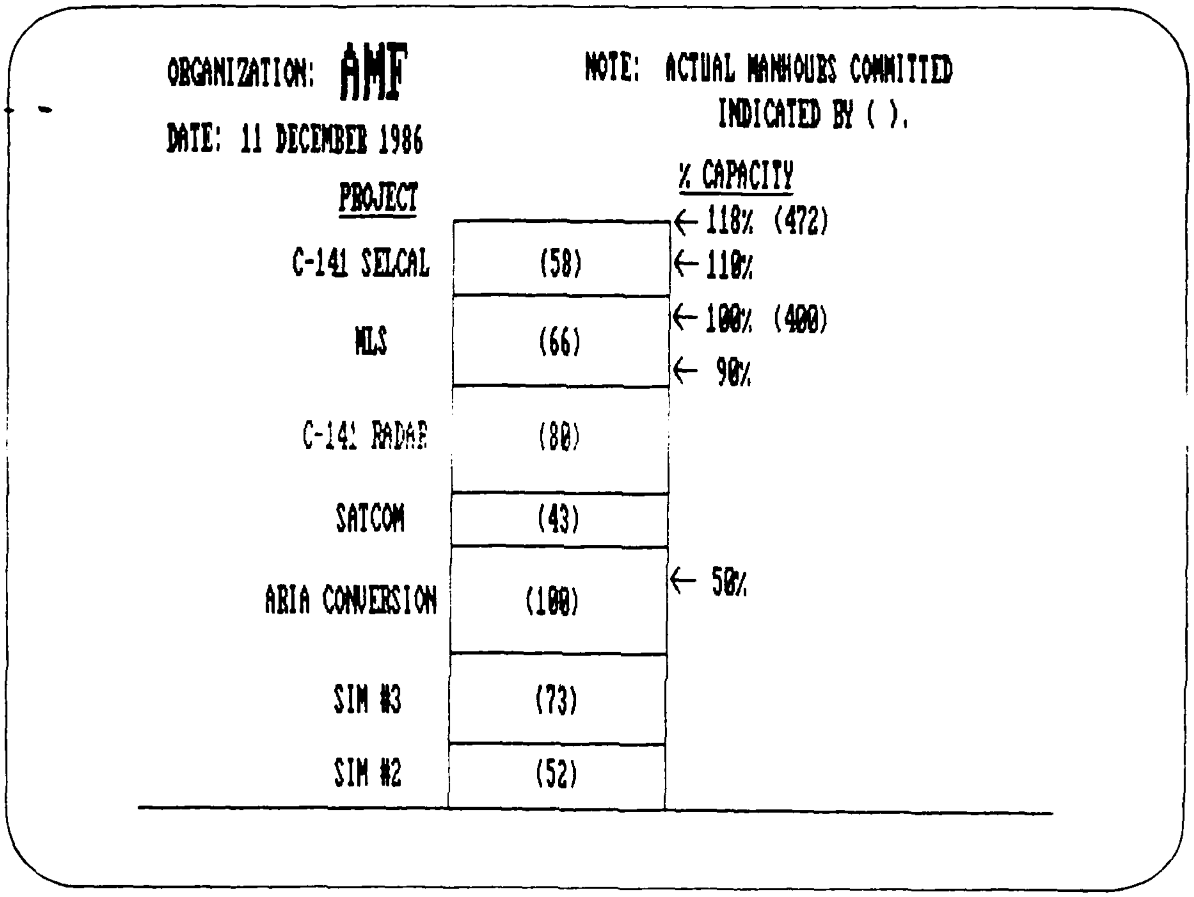



Figure 4.6 Comparison of Project Manhour Commitments

Figure 4.6 further refines the relationships between competing projects by showing the proportionate amounts of manhours required by each project.

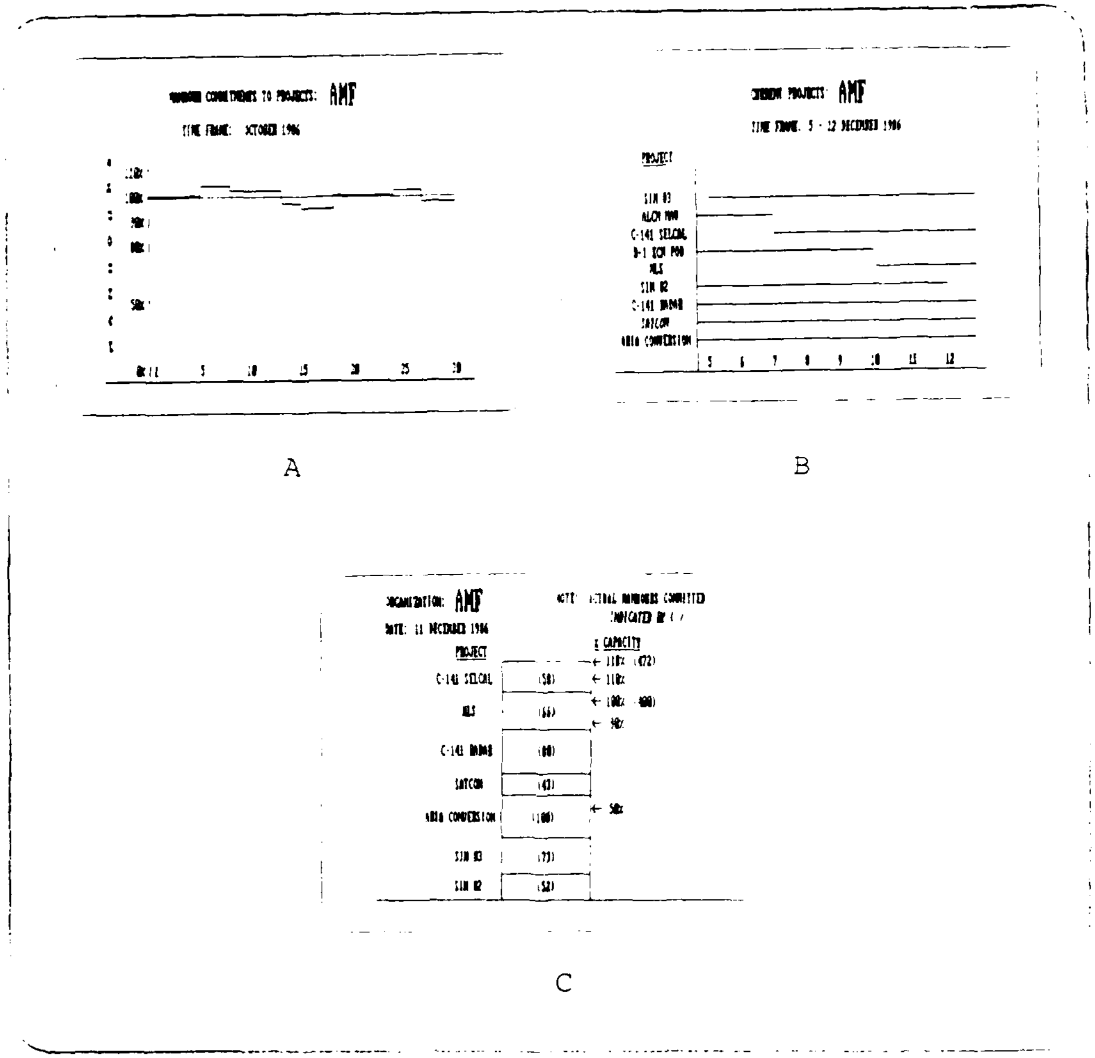



Figure 4.7 Scenario One

These information gathering representations provide a means by which a manager can investigate manhour commitments and identify possible problem areas. One scenario might have a manager looking at the projected manhour utilization levels for a specific shop (Figure 4.7A) to identify periods where commitments exceed the desired level. Having found a time frame where manhour commitments are too high, the manager further investigates to find which projects are competing for the same manhour resources (Figure 4.7B) and how much effort is projected toward each of the projects (Figure 4.7C).

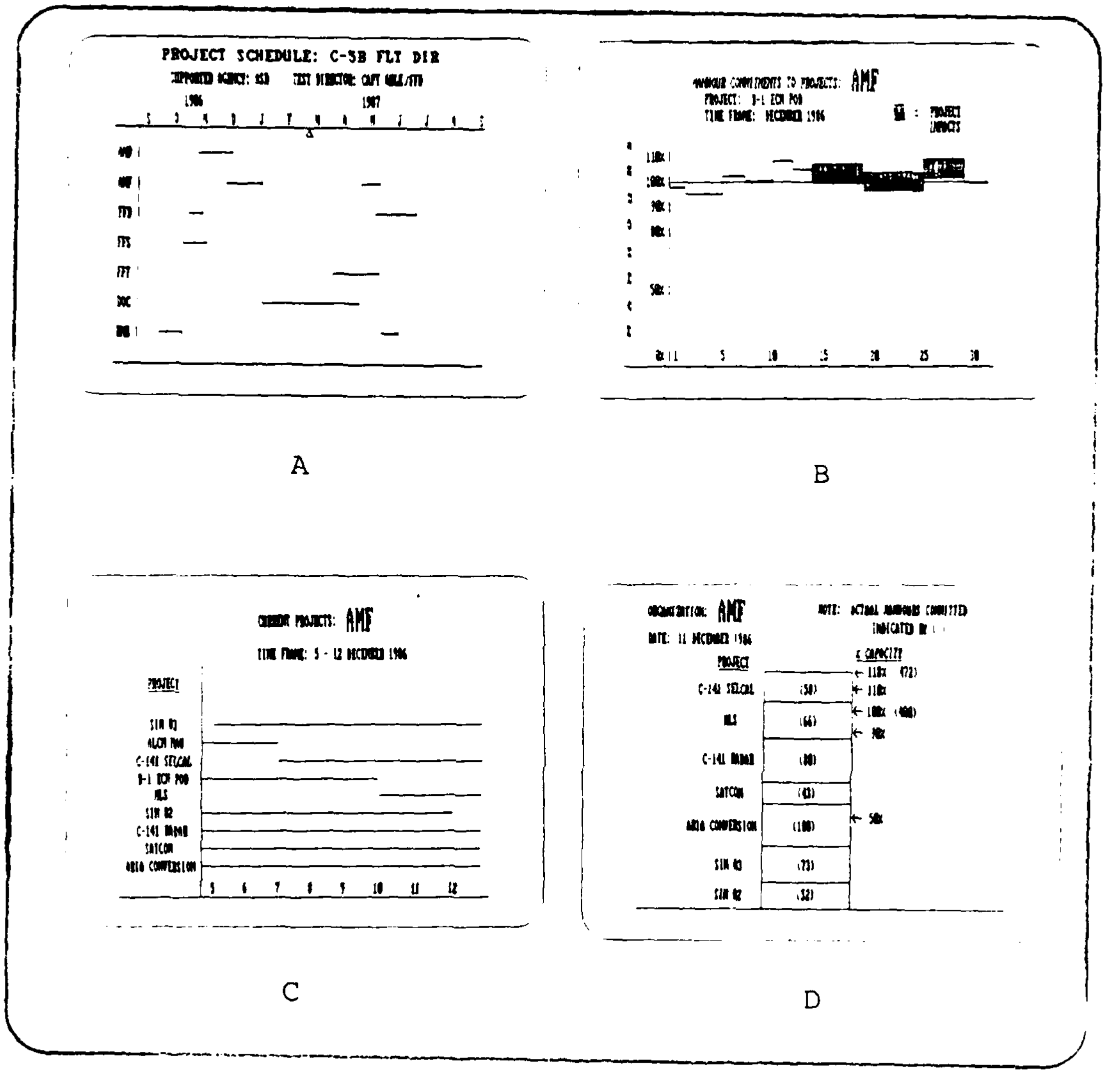



Figure 4.8 Scenario Two

Another scenario might have a manager trying to work a new project into the overall shop schedule. Given the proposed schedule for the project (Figure 4.8A), The manager can observe the impacts of that project on manhour resources for that shop (Figure 4.8B) to determine if the manhour commitment with the new project added is at an acceptable level. If not, he can then identify and investigate the other projects competing for the same manhour resources (Figures 4.8C and 4.8D).

Information gathering representations are tools that can help managers discover and investigate possible problem areas. Although they do not provide solutions to the problems they help identify, they do effectively lead to the next step in the decision process: generating viable alternative courses of action.

Alternative Generation Representations. To effectively address manhour problems, a manager must be able to investigate possible alternative courses of action. The representations shown in Figures 4.9 and 4.10 provide a means to this objective. By enabling a manager to make and record reasonable changes to operative variables of competing projects, these representations initiate the discovery and exploration of viable alternatives.

VARIABLE OPTIONS: RUN # 1

| PROJECT | MODIFICATION / ALT | DUE DATE / SLIP | TEST OBJ / ALT |
|---|---|---|---|
| C-5B FLT DIR | AMF / AMX,CONTRACT | 1 SEP 86 / +30 | 6 / 5,4 |
| C-141 SELCAL | AMF / CONTRACT | 31 MAR 87 | 12 |
| MLS | AMF | 15 MAY 87 / +45 | 8 / 7,6 |
| C-141 RADAR | AMF / CONTRACT | 1 FEB 87 | 15 |
| SATCOM | AMF / AMX | 31 NOV 86 | 6 |
| ARIA CONVER | AMF | 30 NOV 88 | 18 |
| NOTE: SCHEDULED PLANS PRECEDE "/"; POSSIBLE ALTERNATIVES FOLLOW "/". | | | |

Figure 4.9 Operative Variable Selection List

Figure 4.9 provides information about projects that are competing for the same resources. Specifically, it shows the operative variables associated with the competing projects: who will perform the necessary modifications, when the project is scheduled to be complete and the number of test objectives the project entails. In addition to showing the current values for these variables (entries before the "/"), the representation gives known alternative values (entries following the "/"). By selecting a change (for instance slipping a due date), the manager can generate an alternate course of action to compare to the original conditions. Subsequent changes in variables create additional alternatives that can be distinguished numerically from the other alternatives by an assigned run number (such as "RUN # 1").

## SUMMARY OF OPTIONS

| PARAMETER | RUN 1 | RUN 2 | RUN 3 | RUN 4 | RUN 5 | RUN 6 |
|---|---|---|---|---|---|---|
| MOD ALTERNATIVES | | | | C-141 SEL: CONTRACT | C-141 SEL: CONTRACT | |
| DUE DATES | | C-5B FLT DIR: +30 | MLS: +45 | | MLS: +45 | |
| TEST OBJECTIVES | C-5B FLT DIR: 5 | | | | | |

Figure 4.10 Summary of Operative Variable Selections

Figure 4.10 shows a compilation of all runs selected with the specific operative variable changes made for each run. In this manner, a manager can identify and keep track of specific changes in competing projects that he would like to investigate.

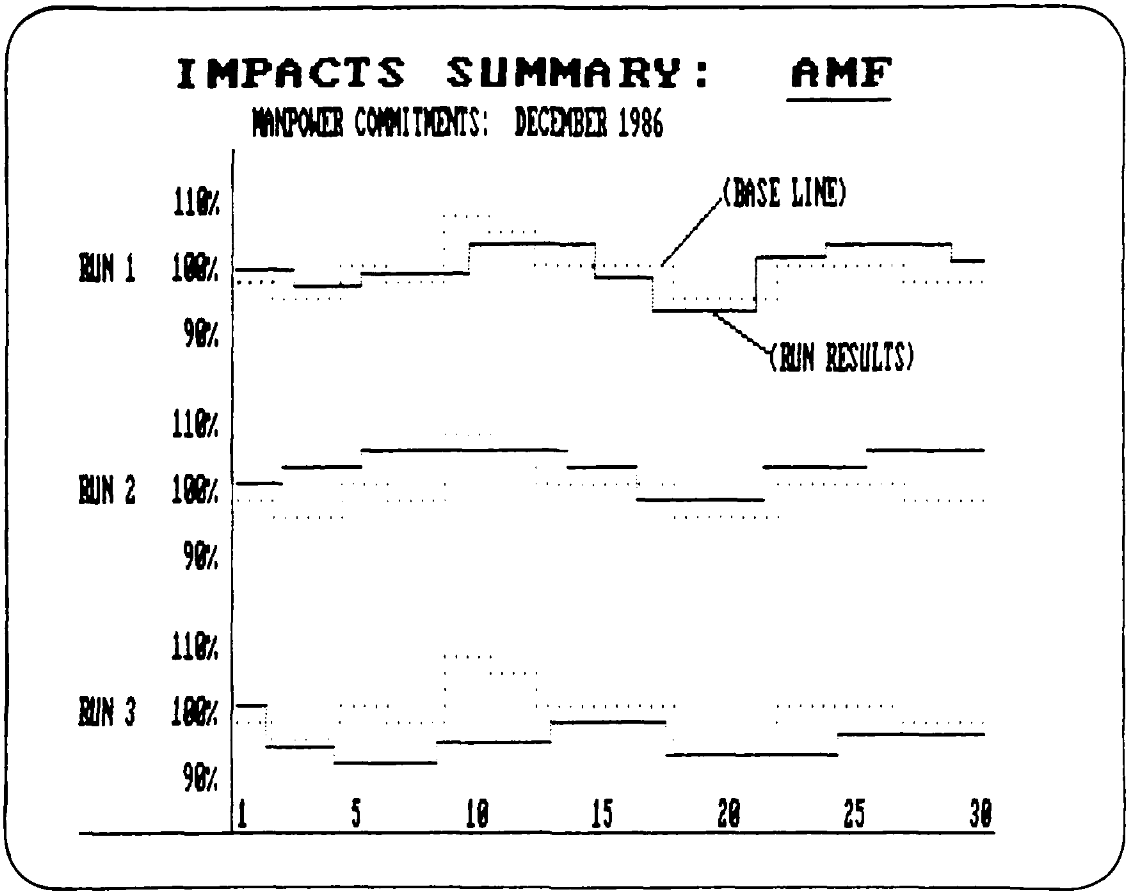



Figure 4.11 Comparison of Alternatives for a Shop

Choice Representations. The final step in the decision process is to choose among alternatives. This requires a means to compare alternative courses of action. Figures 4.11 and 4.12 provide such a means.

Figure 4.11 shows how the alternatives (as identified by run numbers) match up against the base line conditions (the original project schedules before any changes have been made or a selected alternative schedule that has replaced the original schedule as the base line). This graphical depiction enables the manager to directly compare the established base line and alternatives being investigated.

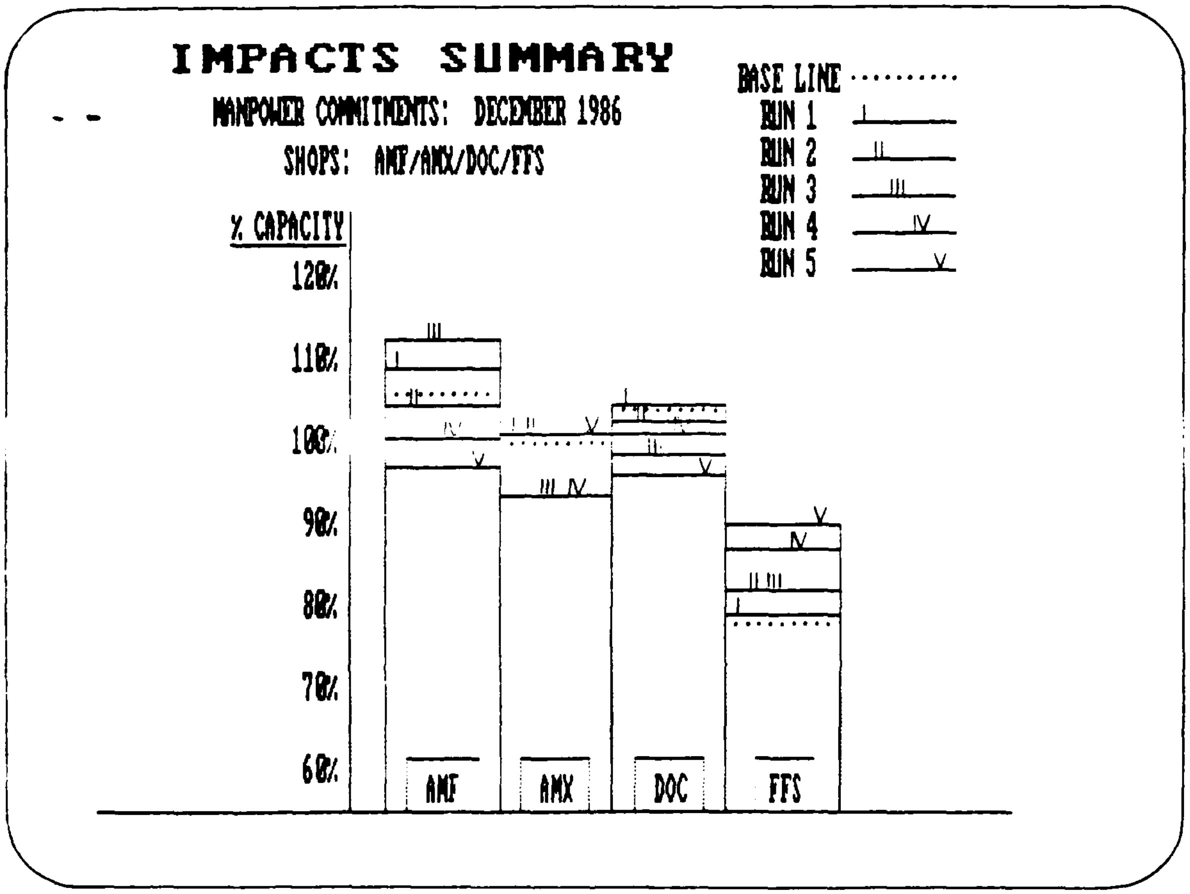



Figure 4.12 Comparison of Alternatives Over Several Shops

In much the same way that Figure 4.11 enables the comparison of alternatives within a shop, Figure 4.12 allows alternatives to be compared on the basis of their impacts over several shops. Through use of separate columns for the selected shops and individual lines for each run (identified by roman numerals), the results of alternatives can be directly compared to each other and to the base line (indicated by a dotted line) from an overall wing perspective. Thus, a manager can assess the impacts of alternatives that might be desirable from the perspective of one shop on the manhour resources of other shops.

Summary of Dialog Representations. The representations presented here are examples of decision support system output that could be used to address the test wing manhour issue. They are only examples. The formats used were devised by the authors in an attempt to display pertinent relationships that have direct bearing on manhour utilization. In keeping with the spirit of iterative building of effective DSS, these representations can and should be modified as needed. There are, however, some factors to be considered when adding new representations. Consistency in the layout of the representations should be maintained so that the user can transition easily among representations. Also, the data required to produce new representations must be available and properly maintained in the data base. The main points to consider are the needs of the users. A DSS can be effective only when it provides its users with the information they need to make effective decisions.

Operations Applied to the Dialog

Screen Layout. So far, only the representations relating to the decision process have been introduced. They have been shown as if the entire display screen were available for their use. However, both Operations and Memory aids (covered in the next section) require the use of menus and thus compete for the same screen space. Figure 4.13 shows a possible screen display layout that will satisfy the requirements of this decision support system.

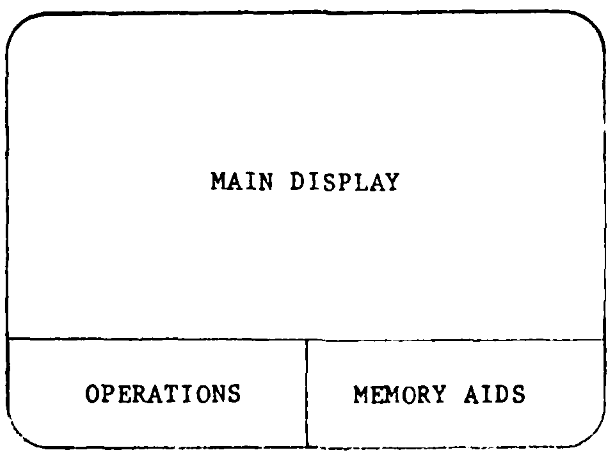



Figure 4.13 Screen Display Layout

Operations Main Menu. Operations are the means of processing and converting decision related data into meaningful results. They are the tools available to managers enabling them to manipulate project and shop information into useful relationships that support the decision process of gathering information, generating alternative courses of action and selecting among them. From the standpoint of DSS dialog, operations can be the menu selections that allow managers to call upon appropriate models [discussed later under "Model Base"] that convert data and information into meaningful representations.

The representations presented in the previous section directly support the project management and scheduling efforts of test wing managers. Thus, the dialog operations (menus) required by the DSS should enable the user to easily reach the desired representations. The following list of menu selections achieves this by reflecting the available representations:

1. Project schedule.
2. Manhour utilization comparison between shops.
3. Shop manhour commitments.
4. Particular project impacts on shop manhour commitments.
5. Projects competing for the same manhour resources.
6. Breakdown of shop manhour commitments by project.
7. Operative variable selection list.
8. Summary of variables selected by run number.
9. Comparison of runs for a shop.
10. Comparison of runs for several shops.

Operations Sub-menus. Because each representation is unique in terms of the information displayed, each menu selection requires specific user inputs (shop designation, time frame specification, and project identification). Thus, once a menu selection is made, the DSS must query the user to obtain the inputs required to perform the needed operations and produce the desired representations. This query process can be accomplished through a series of "sub-menus." A sub-menu would appear automatically after a main menu selection has been made and would enable the user to enter necessary inputs or to check previous entries if a series of representations involve the same project, shop or time frame. For example, a sub-menu for main menu selection 3 (Shop manhour commitments) might be

Enter shop identifier:
Enter time frame (Month year):

or

Current shop selection is: <u>AMF</u>
Current time frame selection is: <u>December 1986</u>

Enter "C" to change an entry or "CR" to proceed

Table 4.1 shows the main menu selection list with required sub-menu input requirements.

TABLE 4.1

Operations Menu and Input Requirements

| Main Operations Menu | Sub-menu Input Requirements |
|---|---|
| Information Gathering: | |
| 1. Project schedule | Project identification |
| 2. Manhour utilization comparison between shops | Shop identification<br>Time frame specification |
| 3. Shop manhour commitments | Shop identification<br>Time frame specification |
| 4. Particular project impacts on shop manhour commitments | Project identification<br>Shop identification<br>Time frame specification |
| 5. Projects competing for the same manhour resources | Shop identification<br>Time frame specification |
| 6. Breakdown of shop manhour commitments by projects | Shop identification<br>Time frame specification |
| Alternative Generation: | |
| 7. Operative variable selection list | Shop identification<br>Time frame specification |
| 8. Summary of variables selected by run number | Shop identification<br>Time frame specification |
| Comparison of alternatives: | |
| 9. Comparison of runs for a shop | Shop identification |
| 10. Comparison of runs for several shops | Shop identification |

Memory Aids Applied to the Dialog.

Main Menu. Mechanisms that help DSS users recall information are memory aids. They keep previously derived, decision related information readily available for use in the decision process. For test wing managers, memory aid requirements can be achieved through an automatic feature, that saves all representations generated during a session, and selectively activated recall, delete and note taking capabilities. In much the same fashion as operation, memory aids can be exercised through use of menus placed at the bottom of the screen display [see Figure 4.13]. The following menu will fulfill the initial memory aid needs of test wing managers:

1. Add text to current representation.
2. Recall a previous representation.
3. Delete previous representations.
4. Delete alternative schedule runs.
5. Print representations.

Adding Text. A typical DSS session may generate numerous unique representations. An effective way to retain significant features of specific representations is to enter pertinent remarks directly on the display for later reference. In this manner, key representations with accompanying remarks are kept intact until the user determines they are no longer needed.

Recalling, Printing and Deleting Representations. DSS users must be able to view or print previously derived representations. Equally important, managers must be able to discard previous displays that have been deemed unnecessary. By choosing the menu selection to recall, print or delete a representation, the DSS must respond with a list of previously created displays to choose from.

Deleting Runs. In generating alternatives, numerous changes in operative variables can be investigated. Keeping track of run characteristics and results can pose a significant problem. While the "Summary of Options" representation is a partial solution, too many runs can clutter and add confusion to the representations that compare results. For this reason, managers must be able to discard runs that have been overtaken in importance. Choosing this menu item must produce a listing of previously established runs with operative variable changes to allow selective elimination. Table 4.2 consolidates the main memory aid menu with corresponding DSS responses and required user inputs.

TABLE 4.2

Memory Aid Menu, System Responses and Input Requirements

| <u>Memory Aid Menu</u> | <u>System Response</u> | <u>User Input</u> |
|---|---|---|
| Add text to current representation | Provide space for writing | Keypad text entry |
| Recall a previous representation | List of previous representations | Representation selection |
| Delete previous representations | List of previous representations | Representation selections |
| Delete runs | List of previous runs/operative variable changes | Run selections |

<u>Memory Aids Windows</u>. An effective method of displaying memory aid information is through use of a display window that uses only a portion of the screen and does not totally destroy the repreasentation in the main display. Such a window is shown in Figure 4.14 and can be used to display previous representations, provide writing space for text additions and list previously displayed representations and alternative schedule runs. Figure 4.15 demonstrates the use of the memory aid window to display previously derived information relating to the primary screen display.

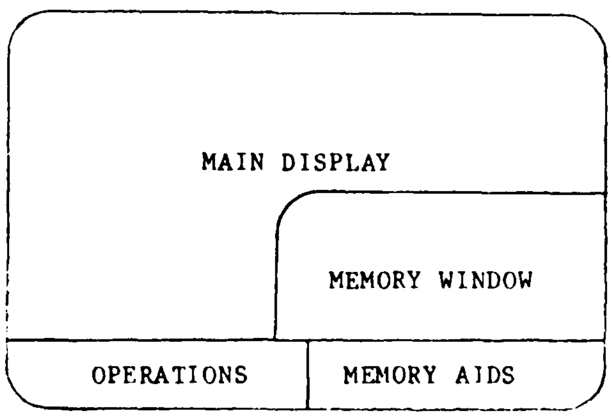



Figure 4.14 Memory Aids Window

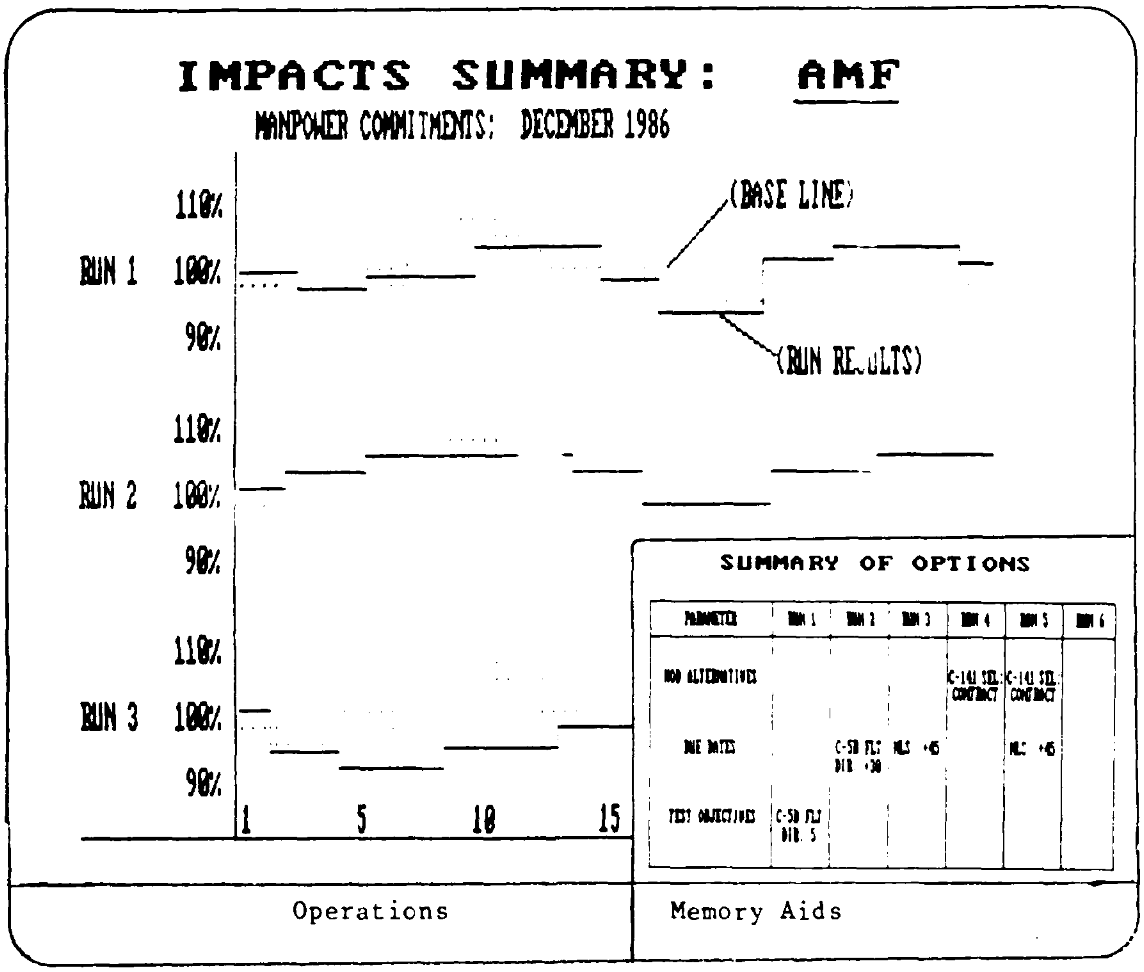



Figure 4.15 Memory Aid Window Example

Control Mechanisms Applied to the Dialog. Control mechanisms provide the direct link between the user and the decision support system. They can be in any form (functions keys, text entries, menu selection numbers, control "windows") that facilitates user control of the DSS. Regardless of form, their primary focus is to enable fast and easy selection of operations and memory aids to support the decision process.

Control Windows. An effective mechanism for directing the test wing DSS is the control "window." The control window outlines and illuminates one possible selection entry at a time. In a set of possible choices, the window initially resides over and highlights a single option. Through use of "arrow" keys, the window can be moved from one item to another until the desired selection is identified and activated by pressing the carriage return.

Control of Operations and Memory Aids. The decision support system must provide easy access to the menus associated with operations and memory aids. Using the screen display layout shown in Figure 4.14, the control window would highlight either "Operations" or "Memory Aids" and entering a carriage return would display the appropriate main menu [see tables 4.1 and 4.2]. Following a selection from either main menu, the corresponding sub-menu would appear with appropriate lists, writing space or prompts for user entries. With this system of menus and control window activations, all operation and memory aid capabilities of the DSS can be exercised with minimal training investment on the part of the user.

Error Messages. Control mechanisms might also include warnings provided automatically by the DSS when the system cannot perform a desired funtion. Examples are errors in user inputs (such as time frames that are out of range) and insufficient memory space for saving desired representations. Whenever the DSS is incapable of accomplishing a required operation, the user must be notified in an understandable fashion.

Control Mechanisms Summary. Menus activated by control windows comprise only one method of directing the DSS. Functions keys or text entries can accomplish the same thing; however, they might require additional user training. The prime consideration is to keep the system as usable as possible without sacrificing flexibility. Employing the same keys that are common in current office equipment and using "function" keys for high use operations are examples of possible system features.

MODEL BASE

The model base is the workhorse of the DSS. It is the middle man between the data collected by the organization and the dialog interface with the end user and decision maker. The model base houses the models and data manipulation programs to support the decision maker's dialog interface; thus, in terms of DSS design, the requirements of the model base are determined by the dialog to be supported. For that reason, the design of the kernel model base for the 4950th Test Wing is presented in terms of the ROMC of the supported dialog.

Representations Applied to the Model Base. The DSS must have a model capable of creating the graphic representations required for the system dialog. The graphics model would support the displays discussed in the previous section of this chapter and control the screen formatting, layout, screen layout, etc. To allow for updating and expansion, the ideal graphics model should allow any two variables to be plotted against each other. The main distinction between DSS and other, more traditional forms of decision aids is in the graphical comparison of alternatives. Thus the graphics model must be able to access data simultaneously for several alternatives and create overlayed representations as depicted in Figures 4.11 and 4.12.

Operations Applied to the Model Base. The model base must support all operations allowed of the user. Besides the creation of graphic representations, the backbone of the operations model base is the scheduling model. The scheduler must take as input the current state of the wing and any changes to operative variables assigned by the user (see Figure 4.9). The scheduler must then generate a schedule and provide data for the graphics model to create representations for this new alternative. On initial start-up, the scheduler must be able to access the external or main data base and create an initial baseline schedule for comparisons.

Capabilities Affecting Model Accuracy. Several capabilities can affect the accuracy and complexity of the scheduling model. As discussed in Chapter II, the scheduler should use some type of heuristic to prioritize activities and then assign them for work (schedule) until resources become unavailable. The selection of a priority rule will affect how accurately the resulting schedules reflect the actual preferences of the wing. To be totally accurate, the scheduler must allow recursion: the ability to schedule a project for initial modification, to baseline flight testing, back through additional modification, further flight testing, etc. It should be able to account for actual work rate distributions: a project requiring 100 hours in 10 days may need a uniform distribution of labor with 10 hours per day, it may need fewer hours in the first days with more later, resembling a triangular distribution, or it may require some other distribution.

The Decision Making Process. To aid in the decision making process, the model should support several experimental methods. The scheduler should be able to generate new schedules allowing no change, minimal changes, and selective changes to the existing schedule. It should be able to assume unlimited resources in order to schedule a new project for its minimum completion time and to identify periods of extraordinary resource usage, and gradually lower resource levels to identify effects on completion dates.

Memory Aids Applied to the Model Base. The model base must support all dialog memory aids. The dialog memory aids allow the user to review past representations, and to type comments on representations before they are saved to memory. To support these aids,the model base must contain a model capable of saving representations (complete with notes and comments) as they appear on the screen. This memory model must also be able to retrieve the saved representations for later viewing and printing. This retrieval is quite different from the creation of representations from raw data required of the graphics model, but is of no less importance to the DSS.

Control Mechanisms Applied to the Model Base. The command language interpreter must be able to recognize any control characters, function keys, or word commands entered by the user and invoke the proper model with the proper parameters to perform the requested functions. The interpreter must ensure the proper access of data by the models. Additionally, it must provide the user with appropriate error messages and prompts for inputs. The listing of such messages and prompts is beyond the scope of this effort and should be accomplished through the iterative implementation strategy discussed in Chapter V.

## DATA BASE

As a storage area for useable data, the data base of the DSS must allow all models to access appropriate memory locations. The selection by the 4950th Test Wing of the Oracle database management system limits considerably the scope to which this work must analyze this aspect of the design of the data base of the DSS. Suffice it to note that the models required to support the user's decision making must gather their data from readily available sources, most notably the wing MIS. Options for access to the MIS data base are discussed in Chapter V. The BSP study has already identified information flows within the organization, and attempts are being made to ensure adequate access to required data in conjunction with appropriate security to avoid misuse of information. Beyond the need for access, the remainder of this work identifies the data base as essentially synonymous with memory space. In general, the data base must allow sufficient memory space for all operations and their resultant data and representations.

Representations Applied to the Data Base. To create new graphic representations, the graphics model must access the schedule data generated by the scheduler model. The graphics model will extract the required data into a separate space and convert the data into the scheduler model format. The graphics model will extract the required data into a separate space and convert the data into the format required for the screen representation. In addition to the raw data for the creation of lines, the graphics model will require screen formats for each type of representation which might be created. The choice representations depict data from several alternative schedules. To present the diversity of information, the graphics model will require memory space for temporary storage of comparison data before transferring data into the screen graphics representation. Finally, the graphics model will require space for the acceptance of input parameters and data directly from the user.

Operations Applied to the Data Base. The initial operation will be an access of the external MIS database, transferring schedule related data into the main DSS data base. To generate alternative schedules, the schedule models will use the main data base, a data base for the new project under consideration, and a data base storing alternative generation inputs from the user. After generation, the schedule model must place the schedule data into an individual data space for each alternative schedule to allow access by the graphics model. Since each of these schedule bases may be used for numerous representations, they should be retained (in the data base) until specifically deleted after the user determines that a given alternative will no longer be considered.

Memory Aids Applied to the Data Base. The data space required for memory aids has the potential for being the largest part of the DSS data base. Because of the desires to place notes on individual representations and access previous representations during the decision making process, memory space must be available to store each viewed representation as a stationary picture, not as raw data. The DSS must have some type of data management to allow labeling or coding of representations such that the user may easily access previous representations.

Control Mechanisms Applied to the Data Base. The greatest part of data base control must be in coding data for easy future access. The scheduler model will create a new schedule base for each alternative schedule. These bases must each be accessible by the graphics model. The user is allowed to save representations for future access. Each of these representations must be properly filed to ensure accessibility. It should be easy to see that any amount of memory space allocated might quickly be overrun by the generation of multiple schedules with several representations each. Thus, understandable error messages must be developed to alert the user to impending memory space depletion, and should guide the user through the steps necessary to select schedule bases and saved representations for deletion. After deletions have been made, the data controller must 'pack' the remaining space and allow for recovery of the freed memory space for future schedule bases and representations.

Summary of the Kernel DSS Design

The decision support system presented in this chapter centers around the series of representations that directly support the efforts of test wing managers to address project management and scheduling problems with respect to limitations in manhour availability. Through the framework of specifically identifying the requirements of the representations, operations, memory aids and control mechanisms, the DSS components of dialog, model base and data base have been defined. Idendifying these requirements, however, is only the first stage in establishing an effective DSS. Careful, thorough implementation followed by continual evaluation and change are every bit as important as the specification of the initial DSS requirements. Chapter V presents the key concepts of DSS implementation and evaluation as they relate to the problem environment of the 4950th Test Wing.

## V. Implementation and Evaluation

### Introduction

This chapter outlines how the 4950th Test Wing might implement the kernel DSS described in Chapter IV. Before directly approaching implementation, the iterative design process is reviewed with particular emphasis on the need for beginning with a small, workable system (the kernel) while including room for eventual expansion. The implementation discussions center on options: those features which do not impact directly on the capability of the DSS to support decision making but affect the usability, expandability, and accuracy of the system. The options desired by the wing will determine in a large amount the software and hardware required for implementation of the DSS. Besides considering the impact of computer options on the kernel system, the wing must be concerned with the impact of people on the kernel system and, conversely, the impacts of the system on the people in the organization. Finally, as a major step in the iterative design process, several techniques for system evaluation are presented, followed by a projection of likely directions for expansion of the kernel.

### Review of the Iterative Design Process

The philosophy of iterative design recognizes two major factors of problem solving: big problems can rarely be solved as a whole, and even the best conceived solutions rarely work optimally on the first try. To avoid becoming bogged down in massive solution attempts, the iterative design approach first selects a small, workable subproblem. This kernel subproblem should be simple enough to be readily solved, yet comprehensive enough that its solution aids in addressing the overall problem. From the kernel system, the users have a basis for recommending improvements and expansion toward solving larger related problems. Improvements can be made, resulting in a new basis for further expansion, and so on. For the 4950th Test Wing, the first step in this iterative process was the design of a kernel DSS presented in Chapter IV. The second step is in selecting the options for implementing the kernel system.

## Consideration of Options for the Kernel DSS

General. Chapter IV contained the essentials of a kernel DSS for project management and scheduling in the 4950th Test Wing. These identified the specific requirements for each of the components comprising a DSS. There exist, however, a variety of methods for actually implementing these basics. Following is a discussion of some of the options available presented by component (dialog, model base, data base).

Dialog. The dialog is the interface between the desicion maker and the machinery of the DSS. As discussed in Chapter IV, the ROMC of the dialog are all directed toward supporting the user's decision making process. The essentials of the ROMC described in Chapter IV define what the dialog should be able to do for the decison maker; but, there are several options for defining how the ROMC look and act. To aid the decision maker in maintaining his train of thought in the decision making process, these options must be implemented with consistency both within the DSS and with other computer systems and programs in the organization. Consistency means that the user will always know what to expect from the system, no matter where in the decision making process he may be. The decision maker should know where to look to find information on the screen and how to input commands, and not be distracted by the DSS machinery. The two main areas available for these implementation options are the screen display and the control structures which can be divided into the six specific items that follow.

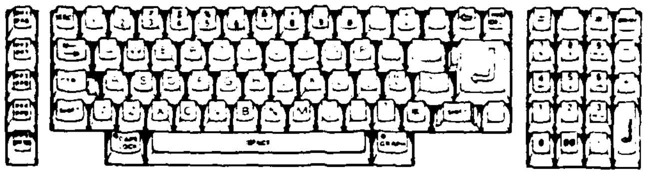
Formatting and Placement. The representations presented in Chapter IV point out the information required by a decision maker in the decision making process, but they do not prescribe the placement of titles, axes, or explanatory and administrative remarks. As noted above, placement of information on the screen should be consistent so the user always knows where to look to find any desired bits of information and is not distracted by a clutter of administrative notes.

Color. Color may be used to enhance screen displays by highlighting important bits of information while keeping routine administrative information unobtrusively displayed, and by allowing overlaid displays for presenting a greater number of comparisons. Shadings and patterns may also be useful if color is not available. Consistency in colors and shading patterns can greatly aid the decision maker in quickly locating desired information on the screen.

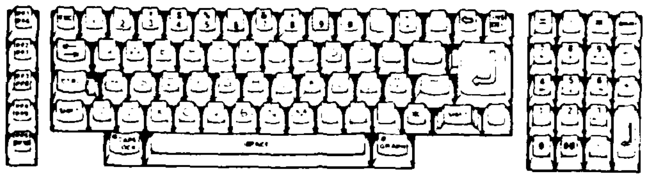
Transitions Between Representations. The method of transition between successive representations can aid or distract the user. Allowing concurrent display of multiple representations and menus (windowing) as a memory aid function can greatly aid the decision maker in making comparisons and in designing additional alternatives, while drawing a full screen for each new display can cause the loss of the decision maker's train of thought. Of course, this feature must be balanced with the need for the greater resolution and the ability to include more information in full screen displays. Whichever method is selected, consistency in the method and in the placement of windows is important for keeping the decision maker's attention on the problem and not on the machinery of the DSS.

Input Mechanisms. The user must be able to tell the system what to do. As depicted in Figure 5.1, common methods include control codes, function keys, text commands, cursor highlighting with a mouse or arrow keys, and light pens. Whatever method is selected, an important consideration is integration with other existing computer programs and systems in the organization. The mechanisms should not compete with responses the user has learned in other areas. As an example, the commonly used word processing program Wordstar (by MicroPro) uses the control-Y code to delete a line. If the DSS uses a control-Y to save a representation, the DSS user might easily find himself deleting lines from a Wordstar file instead of saving his text. If competition cannot be avoided, a completely different mechanism should be used, for instance using the arrow keys or a "mouse" to position a highlighted cursor over the desired command.

Menu Flow. In many instances, the system might require a series of responses to focus the user's desires. For example, when the user wants to see a previously saved representation, the user must first call up the Memory Aids menu, then select the Recall option, and finally indicate the particular representation desired. These flows should be



CONTROL CODES: Example: To print, enter "control P"



FUNCTION KEYS: Example: To print, enter "PF3"

Menu

1. Print 2. Save 3. Delete

TEXT COMMANDS: To make a selection, enter the applicable number at the prompt followed by a carriage return

Menu

Print Save Delete

KEY HIGHLIGHTING WINDOW: Use arrow keys or a mouse to position the window, then enter a carriage return

Figure 5.1 Examples of Input Mechanisms

designed to avoid distracting the user's train of thought. They should allow the user to backtrack gracefully if a wrong option was selected. More importantly, they should allow the user to ask for help or a directory of options. Additionally, the access to the menus should be consistent throughout the DSS, as discussed above, since it could be distracting to use text commands on one menu followed by cursor positioning on the next.

Error Messages. To be useful, error messages must be understandable to the user without being verbose, trite, or patronizing. Additionally, the system should be able to lead the user through any procedures required to correct the problem. For a good discussion of this aspect of user friendliness, implementers should refer to the short, 2-page editorial by Ken Meyer and Mike Harper leading off the March 1984 issue of MIS Quarterly (Meyer, 1984: 1).

Model. The model component of a DSS must support the implementation decisions for the dialog component. As discussed in Chapter IV, the basic dialog requires at least four supporting models: a graphics model capable of supporting all the dialog requirements, a scheduling model capable of generating the required data for the graphics, a screen saver and retriever, and an overall operating system capable of accessing required data bases, interpreting user inputs for model execution, and possibly for reformatting data between schedule output and graphics input. These requirements are determined by the kernel dialog. Since only the kernel DSS is being implemented, there is only a limited set of options to be discussed. These options involve the choice of operating level: should the models operate on the main frame computer or on a microcomputer, and how closely should the models represent the actual state of the organization?

Main Frame Versus Microcomputer. The consideration of placement of the model base on either the wing main frame computer or on office level microcomputers involves costs, speed, flexibility, accessibility, and expandability. Hundreds of project management programs are available for IBM PC level microcomputers, frequently with built in graphics packages (Filley, 1986). The wing must trade off the advantages of lower cost and maintenance of microcomputer programs with the smaller capabilities and expandability of those programs.

Depending the number of main frame terminals and microcomputers available and the amount of computer time dedicated to other, non-DSS uses, either main frame or microcomputer operation might be quicker and more readily accessible. The major drawback to microcomputer operation found in this study is the difficulty in locating a program with sufficient capabilities to accurately represent the day to day operations of the wing.

Model Accuracy and Complexity. The wing has several options in determining the ability of the models, particularly the scheduling model, to accurately represent activities within the wing. Commensurate with increases in accuracy, however, are increases in the size and complexity of the models. The wing must decide what constitutes sufficient accuracy for the type of decisions being made with the DSS. The wing may or may not wish to include indirect activities into the scheduling problem: New Thrust and ARIA activities as they affect available manpower, maintenance, hangar space limitations, and administrative and support requirements of projects. Additionally, the wing must decide how accurately to assign work rate distributions: does a 100 manhour job requiring 10 days mean 10 hours per day, or is the work rate distributed more heavily toward the end of the period? The former assumption may require only a simple calculation by the model to set manpower utilization, while a completely accurate distribution might require the use of the specific manpower availability levels of each shop for each day. Models can be made to use assumed distributions, planned distributions, or estimates based on computer interpretation of historical work patterns. As with the inclusion of extra wing activities, the more accurate the distribution assumptions in the model, the more complex the model will become. The increase in complexity will increase the time required to produce a schedule and increase the data storage space required to support the model, in addition to increasing the accuracy of the end result.

Data Storage. As with the model base, the requirements for the data base are determined by the dialog and model bases. The data base must have enough memory space to support the model and dialog needs. Implementation options center on where the memory space is physically located, how the space is used and how much space is allowed for expansion through the iterative design and implementation of the DSS.

Where the Data is Stored. The choice of where to store the DSS generated data is on the wing MIS main frame computer or on a local microcomputer. The wing has contracted for installation of the Oracle data base management system as the basis for the overall MIS, as described in Chapter I. Oracle will hold all the primary project management data: dates, milestones, costs, resource availability and utilization estimates, etc. While Oracle will hold all the initial data required for the DSS to generate basic schedules, the wing has the option of storing the DSS generated data (schedule bases, graphics data, etc.) on Oracle or of treating the Oracle system as a wholly external data base while maintaining the DSS generated data locally on a microcomputer. In making this decision, the wing should consider the availability of adequate memory space on each system, the ease of model access to the data, security of the original data from unauthorized use or accidental changes, and the ability to update the main data base after decisions are made.

Data Save Options. In creating alternatives for comparison, the user will selectively alter specific items in the data base and view several representations of the effects. The data associated with each alternative must be maintained identifiably separate to allow the user to recall representations and generate new representations from previous alternative runs. Thus, the volume of data generated during a decision making session is potentially huge. The wing has the option of controlling how this volume of data is saved: automatically or selectively. Automatic saving would relieve the decision maker of having to interrupt his train of thought and consciously activate the save process; however, the price of this convenience is the space required to save a great deal of potentially unnecessary data. Selective saving would require the user to actively invoke the saving process for those data bases and representations he believes he will want to use again. Since the decision maker cannot foresee the future, he may not save items he later desires, requiring a rerunning of the schedule and graphics models. In either case, the user should be able to easily remove unwanted data, freeing space for later alternatives.

Insuring Room for Expansion During Design Iterations. As discussed in several places, the kernel system designed in this effort is only a start and would be expanded to include more comprehensive project management and scheduling problems. The wing must trade off the costs of installing more data space now than is currently needed with those of installing a small data base which might require replacement to allow expansion. If memory space is at a premium, the system could be designed to allow only hard copy saves of representations to a printer instead of saving to memory. Additionally, the data bases created by the scheduler could be overwritten when a new alternative is generated. Each of these memory space saving options has a price, however, in not allowing the user to make on screen comparisons of previous representations without rerunning the schedule and graphics models.

## Involving the People of the Organization

General. While the implementation options discussed above will determine how the kernel DSS looks and acts, the people in the organization will determine how the system is used. Implementers must consider the impacts of perceptions on the acceptance and proper operation of the system, and the impact of the system on the work habits of the people. In considering these impacts, this study divides the people associated with the DSS into three categories: data inputers, DSS users, and system overseers. The inputers are the grass roots level people who will be responsible for ensuring the accuracy and currency of the main Oracle data base. The users are the commander, managers and test directors who will be accessing the DSS to help investigate and solve project management and scheduling problems. The overseers are the technical experts responsible for insuring accurate and timely data input, educating the users, and monitoring the machinery of the DSS. Each group of people will have special needs and requirements to be fulfilled for the DSS to be able to help in project management decision making.

Data Inputers. The DSS is designed under the assumption that data is available and accurate. The inputers are the source of that data and so are a vital part of the overall DSS operation. The implementers of the kernel system must insure that the inputers fully understand the importance of timely and accurate data input into the main Oracle data base. Resistance to these input requirements could arise in several areas. From a review of initial design documents for the wing information system, it is evident that the wing intends to save much more data in its main Oracle data base than is now required by hand. This increase in requirements will impose a higher workload on inputers, especially for non-typists. This workload may be eased by insuring easy access to terminals and by developing simple procedures for inputting data, updating data, and correcting typing errors. Balancing the ease of access, however, must be a security system to avoid accidental destruction of the main data base and unauthorized access to sensitive data. Long range decisions will be made based on the data stored in the main data base and the full understanding and cooperation of the inputers is the key to ensuring the accuracy of this basic data. As always, the basic law of data processing holds true: Garbage in - Garbage out.

Users. The commander, managers, and test directors will use the DSS to investigate and solve project management and scheduling problems within the wing resulting in monetary and service obligations to their customers. Because of the importance of the decisions being made, these users will not adopt the system unless they are confident of the accuracy of its results. A first step in developing such trust is in understanding how the system works: the assumptions and limitations of the models, the importance of relationships presented in the representations, and the methods used for insuring the currency and accuracy of the beginning data. Steps must be taken to overcome resistance to the technologies advanced by the DSS methodology. In addition to education in the workings of the system, education in the hands on use of the system will aid the users in transitioning from intuitive methods of problem solving.

Overseers. The overseers include the "champion" and technical experts. A champion is an individual who believes that the system must be implemented and used, and who has sufficient influence to insure that end. The champion is essential to overcome the inertial resistance to change inherent in any organization. The technical experts are the human interface between the users and the machinery. Besides ensuring that the hardware and software are operating properly, they are responsible for educating the users and inputers in operating the system properly. The technical experts are also responsible for overseeing the evaluation and iterative design process of the DSS. At this point, it is important to realize there is a difference between equipment oriented experts (for example, data base managers who are generally technicians exposed to the needs of the end users) and application oriented experts (for example, data managers who are users trained in the technical aspects of the information system). In spite of their differing perspectives on the organization and use of the information system, a balance of both equipment and application oriented overseers is important. Together, they help identify the needs and desires of the organization and users, watch for technology and software advances in the marketplace, and match the two. In short, the overseers help insure the success of the DSS by providing the impetus and guidance for implementation, use, and growth.

Recommendation. This study recommends a measured and coordinated implementation centering on the human element of the DSS. One examination of this philosophy has been presented by El Sawy who approaches implementation as a gradual infusion of a new set of values which emphasizes "the coexistence of computers and people" (El Sawy, 1985: 135). He supports the use of an initial core of users with the need for the new technology and the enthusiasm to put the new technology to work on their own problems. This core becomes the grass roots teachers who, through their use of and belief in the new system, attract other workers to accept the technological changes. Whether or not the wing leadership accepts this view of cultural infusion of values, they must consider the effects of any technology advances on the people of the organization, and vice versa.

## Evaluation of the DSS

General. Evaluation is checking to determine if the DSS is helping the decision makers and how it might be expanded to help them more. It is the critical link between successive generations in the iterative design process: the link which determines what the next iteration should include. As described by Sprague and Carlson, the overseers should evaluate the four P's: productivity, process, perceptions, and product (Sprague, 1982: 160). These measures and some suggested techniques for their evaluation are summarized in Figure 5.2.

**PRODUCTIVITY MEASURES** (Impact on Decisions)

1. Time to reach a decision
2. Cost of making a decision
3. Results of the decision
4. Cost of implementing the decision

**PROCESS MEASURES** (Impact on Decision Making)

1. Number of alternatives examined
2. Number of analyses done
3. Number of participants in the decision making
4. Time horizon of the decision
5. Amount of data used
6. Time spent in each phase of decision making
7. *Time lines of the decision*

**PERCEPTION MEASURES** (Impact on Decision Makers)

1. Control of the decision-making process
2. Usefulness of the DSS
3. Ease of use
4. Understanding of the problem
5. Ease of "selling" the decision
6. Conviction that the decision is correct

**PRODUCT MEASURES** (Technical Merits)

1. Response time
2. Availability
3. Mean time to failure
4. Development costs
5. Operating costs
6. Maintenance costs
7. Education costs
8. Data acquisition costs

Figure 5.2 Examples of Measures for DSS Evaluation (Sprague, 1982: 160)

What to Evaluate. Beyond the general evaluation measures discussed by Sprague and Carlson, the wing should address several specific areas in evaluating the kernel system.

Productivity. In terms of evaluation, productivity asks whether the wing better off with the DSS and are the wing goals being better met. If there is no improvement in the accomplishment of the wing goals of more constant manpower usage, more projects completed and completion dates closer to planned, etc., then there is no reason to require the extra costs of maintaining a large computerized scheduling system or the extra effort of collecting all the extra data required by the automated system. If the DSS is not contributing adequately to productivity, the problem may lie in its design or implementaion. Areas to investigate for improvement include the accuracy and comprehensiveness of the models, and the clarity and appropriateness of the representations.

Process. Is the process by which the decision makers reach decisions improved? The primary point for evaluation of the decision making process is whether or not the decision makers are taking advantage of the DSS capabilities for providing more information than would be available manually. The DSS allows decision makers to generate and compare various potential schedules. To take full advantage of the DSS, the decision makers should make "what if" analyses and compare a large number of alternatives covering a range of possible contingencies. If the decision makers only compare two or three likely alternative schedules, they are not tapping the potential of the DSS, and the expense and effort to maintain the DSS may not be warranted. If such is the case, an effort must be made to determine why the decision makers are not taking advantage of the system: distrust of automation, insufficient time or education to use the system properly, psychological fear of non-acceptance of decisions by the organization, inaccuracies in the models, lack of clarity of the representations and the misunderstanding of what they are showing, uncertainty regarding the real goals of the organization, or any number of other reasons. If the DSS is to be worth the time and effort to maintain, it must be used, and it must improve the decision making process.

*Perceptions*. The perceptions of the decision makers and the rest of the organization determine whether or not the decisions based on the DSS will be accepted and implemented. If the people in the organization do not trust the models, either for accuracy or completeness, they will not accept the resultant schedules or decisions. This lack of trust may stem from many of the same roots as problems identified above with the decision making process.

*Product*. An evaluation of the product involves a measure of the actual performance of the DSS itself. Are the benefits of organizational improvements and customer satisfaction gained by use of the DSS worth the expense of developing, operating, and maintaining the DSS, and educating its users? Methods for reducing such costs should be investigated to improve the product, the DSS.

*How to Evaluate*. The users and overseers must actively seek out any problems with the system, the people's response to the system, and the end results of using the system for its intended purpose. Prompt resolution of any problems will help ensure the system actually helps the wing in its attempt to resolve project management and scheduling problems.

*System Evaluation*. Some techniques for finding problem areas within the DSS include surveys and questionnaires of the users, inputers and people affected by the decisions made to determine their use of the system and confidence in its results. If the wing is able to maintain files of schedules and decision making sessions from periods before DSS implementation, comparisons can be made regarding the impact the DSS generated schedules make, the accuracy of DSS schedules (that is, how much they change after being implemented), the time required to make decisions, and the amount of data and number of alternatives used in making decisions to see if they are better than before.

*User Inputs*. The DSS users are an important source of inputs for system improvement. Since the essence of the DSS is user support and interaction, if users believe they could make better decisions with the rearrangement or addition of representations, menu options, or methods of control and data input, the overseers must try to expand the system to provide such additions and changes. Ease of use and the accuracy of resulting decisions are the foundations on which the DSS philosophy is built.

The Valusek Note Card Method for User Inputs (Valusek, 1985). This study has found an easy to use method for gathering user inputs for change: the Valusek Note Card. As shown in Figure 5.3, the Valusek Method simply asks users to jot down on a note card sized form ideas for improvement at the time of the ideas. The importance of on the spot notes cannot be overemphasized. This method recognizes that when asked for suggestions, a person will undoubtedly remember only those ideas which recur with enough annoyance to be ever on his mind, or those which have occurred recently. Thus, just asking for ideas will not insure receipt of the best ones. The notes do not need to be elaborate, just there. Addition of comments relating to the circumstances during which the idea occurred may help the user remember what he really meant, so comment space should be provided. Once the note cards are completed, they should be tossed in a desk drawer and forgotten until the overseers come to collect them. Forgetting about previous notes frees the user from trying to actually solve the problem (the job of the overseers) and allows the user to continue his work and devise new ideas as they arise, even repeating old ideas as frequently as the circumstances giving rise to their inception occur. Dates and labels can be used as sorting keys to help identify trends, seasonal problems, and major areas needing investigation. In this way, the overseers may be able to gain a feel for the relative importance of ideas to determine which problems and expansions to attack first.

| DATE: | LABEL: |
|---|---|
| NEED/IDEA | CIRCUMSTANCES |
| | |
| | |
| | |
| | |
| | |

Figure 5.3 Note Card Inputs for DSS Evaluation (Valusek, 1985)

Summary of the Evaluation Process. Evaluation is an integral part of the iterative design process. As noted above in the review of the iterative process, the system presented in this study is only a kernel to help with one aspect of a much larger problem in project management and scheduling. To take into account all the factors of the larger problem, the system must be expanded through the process of evaluation and iterative design. Additionally, the kernel system presented here may not work completely as intended in itself, and so will need to be evaluated for reliability, accuracy, and acceptance. Since the key to the success of a DSS is usability and interaction with the decision maker, inputs from the users are vital to gain and maintain their acceptance of this new information technology philosophy.

## Foreseeable Expansions and Future Iterations

General. From the kernel system presented in this study, expansion may be made in several directions. Readily identifiable are improvements in the accuracy of the scheduling model, expansion to include additional goals and objectives, and expansion to other levels in the organization.

Improvements in Model Accuracy. As discussed in the section on model options above, the wing may wish to implement a kernel system with a smaller, less comprehensive, but less expensive scheduling model. If that is the case, one of the first areas they may wish to expand is to improve the model to more accurately reflect actual conditions within the wing. These accuracies may be in work rate distributions, the inclusion of indirect wing activities into the schedule, or allowance of a recursion (projects moving from initial modification, to flight testing, back for additional modification, more flying, etc.).

Addition of Objectives. The scheduling decision should actually be based on more than just the level of manhours used to complete all projects, as designed into this kernel system. The wing should consider an alternative schedule's effects on completion dates, reimbursable costs, completion of all desired test objectives, use of resources beyond manpower, use of hangar space and support equipment, utilization of aircraft, and so on.

Expansion to Other Organizational Levels. The kernel system designed in this study centered on wing level decision making with regard to project management and scheduling. This system could be adapted for use at lower levels. In directorates and shops, the system could provide an aid to forming resource and time estimates of potential projects. Additionally, it could aid in internal scheduling of workers, equipment, and supplies at the shop level.

Summary. The kernel presented in this work is only a beginning. The design is only a first step in assisting the wing in making better project management and scheduling decisions. Even better work may be made with a conscious process of evaluation and continuation of the iterative design process.

# VI. General Observations and Comments

## Introduction

This chapter presents to future designers of DSS, a discussion of several potential problems in DSS design and implementation. These concerns are borne from the problems experienced in the design of the specific DSS presented in this study. This study does not discredit any of the MIS efforts of the 4950th Test Wing, to date; indeed, many of the wing efforts toward building a consolidated data base are also steps toward building the data components of future DSS. However, this study has found several problem areas which should be addressed before designing or implementing any information system. One problem is in finding the proper approach to implementing a solution. As will be shown, one must balance the desire for the technologically "perfect" decision aid that will take a long time to implement with the need to help the decision makers now. Expanding from the solution approach is the problem of finding the right kernel; that is, determining how large and comprehensive a system to implement on the first try. No matter the size and scope of the kernel, implementation will undoubtedly meet with organizational inertia. To overcome these problems and aid in gaining access to essential people and data, the need for a true champion within the organization is presented. Finally, problems of timing, personnel turnovers, and budgeting and manning limitations are discussed as they affect information systems in the military.

## Finding the Right Solution Approach

Differing Views on Designing Information Systems. This study found differing points of view regarding how to design and implement an information system. In designing the kernel DSS, the authors held lofty goals for an ideal system: letting the decision requirements drive the system design. The implementers of the wing MIS, on the other hand, oriented their efforts toward existing capabilities in order to develop a system that works now: letting technology drive the system design. Both views have drawbacks, yet both are necessary for the best design and implementation of information systems.

The Problem Oriented View. In the search for the perfect DSS, problem oriented designers are careful to insure they have identified the right problem; then, they let the requirements of the problem drive the requirements of the system. If the problem is too large to solve in one iteration, the designers insure a logical path is available from the selected kernel system to the desired end system. Additionally, designers try to foresee the decision making process and provide every aid to the decision maker, forcing technology to meet their demands for capabilities. Concentrating on the demands of the problem while ignoring the limitations of current technology, however, runs the risk of developing an ideal, yet infeasible system.

The Solution Oriented View. From an organizational, operational perspective, solution oriented implementers tend to search for readily available solutions to problems. While such a process may insure a system is quickly installed, letting current technology drive the solution technique may result in inadequate investigation of the problem and incorrect definition of what the decision maker requires.

The Wing Approach to Information Management. The 4950th Test Wing, as discussed in Chapter I, recognized deficiencies in their handling of internal information. They saw the problem as one of managing the flow of information within the wing and undertook a BSP evaluation. The result of the BSP evaluation was the design of a large central data base in the MIS style of information management.

Results of the Wing Approach. Initially, the wing correctly used the information flow requirements of their problem to define how they would solve the problem. Therefore, as designed, the wing MIS data base should adequately address the original problems with information flow and use. Indeed, the MIS data base is a necessary precursor to successful information system implementation, as it insures the availability of required data. However, the information desires of the wing have expanded and now exceed the capabilities of the data base centered MIS philosophy. While this is not inherently bad, the wing has not adapted their solution approach commensurate with the expansion in desires. The wing is not letting the requirements of each information need determine how they will satisfy the need, as they did with the original information flow problem. Rather, they are attempting to force everything into the MIS framework without determining whether the framework is appropriate: letting their current technology drive their approach to the problems.

An Example of Allowing Technology to Drive Design. The wing desires the ability to perform "what if" analysis on project schedules to determine future requirements and the effects of project delays or variations. They envision a computer model accessing the MIS data base to generate schedules, but they have not considered how they will use the new schedules to determine those requirements or effects, or what they will do once the requirements and effects are found. In allowing the MIS technology to define their approach to the scheduling problem, they are not insuring the needs will be satisfied. They have lost sight of the real needs of the end user.

The Approach of This Study to Information Management. To insure the satisfaction of the user's needs, a distinction must be made between design and implementation while realizing the need for both. Using this philosophy, this study divided design and implementation between two separate chapters (IV and V) to emphasize their distinction and importance. In designing the DSS, this study insured that the requirements of the decision drove the requirements of the DSS design: the essential ingredients of the dialog, model, and data components. In discussing implementation, this study presented a range of alternatives: from optimistic, state of the art and beyond, to simple and available. The strict reliance on decision requirements insured the kernel system would address the problem at hand and would be expandable to incorporate larger portions of the problem. The range of implementation options insured the wing could implement the system at a level the organization would accept and could afford while providing a framework for investigating improvements in the future.

Recommendation. To insure the real needs of the users are satisfied, this study recommends maintenance of a distinction between the design and the implementation of DSS, realizing there must be a balance between the two. First, to insure the system adequately addresses the right problem, design must concentrate on meeting the requirements of the decision process. Next, to insure the system can be put into use, implementation must consider the capabilities and limitations of current technology. Finally, to insure the implemented system aids the decision maker as much as possible, technology advances must continue to be applied toward the "ideal" design goal.

Finding the Right Kernel

General. Chapter III developed the importance of finding a kernel problem small enough to be solvable, yet large enough to be meaningful and to aid in understanding and solving the larger overall problem. Trying to tackle too large a problem can lead to large solutions requiring long lead times and large resource commitments before actual implementation. On the other hand, narrowing the scope of the investigation too much runs the risk of wasting effort and resources on an inconsequential system that cannot be expanded to meet the full problem. A balance must be found.

The Wing Approach to Solution Size. The wing approached their internal information problems through a BSP evaluation of information flows. The result was the design of a large central data base in the MIS style of information management. The wing chose to implement the MIS as a whole. They are simultaneously developing the data base and 26 separate modules to access the data base for the specific information needs identified through the BSP evaluation. By trying to implement the total system in one step, the wing has committed itself to a long term program with extended lead times before anything is available to the users. This has resulted in a build up of expectations regarding how the MIS will help the organization manage its information flows, followed by a decline in enthusiasm from the lack of visible progress (Interviews, 1985).

This Study's Approach to Solution Size. This study began by looking at the subproblem of tactical planning: scheduling and schedule analysis to determine resource, workload, and marketing requirements out to two years from the present. The study narrowed its scope to the problem deciding how to fit new projects into the existing schedule. The study then further narrowed its scope to consider only how manpower limitations affect that project scheduling decision. The goal was to concentrate on a subproblem small enough to be solved with the resources available, yet relating to the overall tactical planning issue such that it would be a valuable aid in tactical planning decisions and could be expanded to encompass the overall problem as time and resources allow. By providing a small but working system, this study hoped to instill in the organization a confidence in the DSS philosophy toward solving information related problems. This confidence would help generate and maintain an enthusiasm toward expanding the initial system and applying the DSS philosophy to other problems.

Recommended Approach to Solution Size. This study recommends a combination of the above approaches to solving large information related problems. Having a master plan can help the organization focus its efforts and maintain a steady course toward solving all of the identified problems. The idea of starting small and expanding, however, has the immediate advantage of providing visible results. A view to combining these two approaches maintains the guidance of a large master plan; but, it replaces the long lead times of massive implementation with the visibility of the iterative design philosophy. With the wing plan containing 26 major problem areas, the combined approach would address the problems sequentially, not simultaneously, and would apply the "start small and expand" approach to each. Once a small system is implemented for a given problem, the overall plan must be periodically evaluated and updated. The choice of whether to expand the system or attack another problem area, then, would depend on the results of the evaluation of the efforts required, the resources available, and the desires of the users in terms of the overall implementation plan.

A Real World Approach to Solution Size. The wing has made sizeable resource and effort commitments to the MIS goal of a massive, integrated data base. They have progressed to the point where it would be very difficult to reorient their implementation philosophy. As a result, the data required to support individual information needs, in particular the DSS designed in this study, will be unavailable in the near future. As an alternative, the wing may want to select a small solvable information problem based on the ability to implement a solution immediately, focusing on infusing DSS concepts and philosophies into the organization while waiting for the data and user backing to be available to attack the real problems. For example, a scheduling problem similar to the tactical planning problem addressed by this study can be found in any of the work force shops. Since the modification center has a requirement for accurate internal scheduling and already maintains the data necessary to support a scheduling DSS, the wing could implement a small system (similar to that outlined in this study) at the modification center level. Since such a system could not be readily expanded to encompass the overall wing perspective, it should not be considered a kernel to the overall wing problem; however, it would help to show the organization how the DSS philosophy of decision aids can be of benefit, and could help foster a desire to implement the wing level system when its supporting data requirements are met.

## Finding the Right Colonel

The Importance of Finding a Champion. A champion is an individual who believes that the system must be implemented and used, and who has sufficient influence to insure that end. He is essential to overcome the inertial resistance to change inherent in any organization. While this study was invited and formally supported by the wing, the DSS design had no real champion from within the organization. As a result, several areas were encountered which limited the speed and depth of investigation of this effort.

Access to Decision Makers. Without a champion, access to the decision makers was limited. This study had to rely on official statements of intent for identifying organizational goals, objectives, and operative variables, and on imaginative designs for developing the representations believed important to the decision making process, confirming them only after the design was nearly complete. A champion could have insured better access to the decision makers, the end users of this effort, which would have gained their active involvement in the iterative design of the dialog component by allowing early confirmation of the kernel system's aim and testing of representations against their expectations and desires.

Access to the Organization. Without a champion, access to the various shops and directorates in the organization was limited. While this study was able to interview key people within certain organizational units with regard to current manual project tracking methods, it was unable to gain adequate access to the two most important groups from the standpoint of system design: the modification center and the test directors. The modification center has in being a semi-automated shop level scheduling system. Their assistance in identifying the decision making process at their level would have been invaluable as a guide to identifying the process at the wing level. Additionally, since the modification center had the procedures and data readily available to support scheduling decisions, they could have helped this study in identifying what representations the decision makers might desire, and served as a test bed for model and representation development. The test directors, being potential users of the scheduling DSS, should have been directly involved in the iterative design of the dialog component to insure the inclusion of desired capabilities. The test directors could also have helped in decisions relating to the accuracy requirements of the models. A champion could have aided in gaining access to these groups, resulting in better user involvement and a better refinement of the kernel design.

Access to Data. Without a champion, access to current scheduling data was nonexistent. In its investigation of the accuracy requirements of potential scheduling models, this study was limited to one small set of four to six year old data. The data presented a picture of project flow through the organization that was far from complete or accurate, resulting in the questionable validity of model tests. When this study requested more current and complete data, of the type required by the coming MIS, the wing was unable to respond. A champion could have instilled in the organization a sense of importance and preparation in being able to provide the data required as input to the planned wing MIS.

The Grass Roots Need for a Champion. Without a champion to overcome organizational inertia, there was no grass roots desire to see new systems implemented or to aid in their design. As found during the course of this study, the lack of a true champion led to delays in design, reliance upon sketchy data, and an inability to actively involve the organization in the iterative design process. While one may be able to design a DSS without a champion's aid as evidenced by this study, implementation would be very difficult and organizational acceptance nearly impossible without the grass roots support generated by a true believer in the needs for and the capabilities of a DSS.

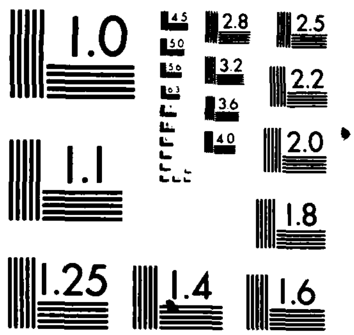
## General Comments on DSS in the Military

Introduction. This section discusses three additional areas that can impinge on the success of DSS efforts. While they are applicable to any organization contemplating .. DSS or other information system, these areas are especially critical when coupled with the unique characteristics of military organizations. The first area regards the time required to fully design and implement a large information system. This time is relatively long as compared with the reassignment rate typical in a military organization and can adversely affect implementation. Second, the rapid changeover of military commanders and decision makers can hinder the use and acceptance of systems already in place. Finally, the budget and manning constraints imposed from outside the organization can impair the ability of the organization to meet the technical requirements of advanced information systems.

Implementation Time. Information systems take a long time to fully implement. This fact is true regardless of the implementation style used, from the total system approach to iterative design. The 4950th Test Wing plans to invest four to six years in the design and implementation of their MIS using the total system, all at once approach. The DSS proposed by this study recognizes that the process of expansion of the kernel system to encompass the full scope of the project management and scheduling problem in the wing will also take years of evaluation and iterative design. The length of time required to fully implement either system may be longer than a normal tour of duty for military personnel, leading to a changeover in the organizational leadership, the project champion, and the grass roots end users. These changeovers can result in the redirection of efforts

AD-A172 448 INVESTIGATION AND DESIGN OF A PROJECT MANAGEMENT DECISION SUPPORT SYSTEM F..(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. 2/2

UNCLASSIFIED R H BLACK ET AL. MAR 86 AFIT/GST/ENS/86M-2 F/G 5/1 NL




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

and the degradation or loss of support at all levels within the organization, severely hampering the successful implementation of the information system.

User Confidence. In addition to contributing to loss of support, personnel changeovers can affect the confidence of the decision makers in the output of the information system. If new commanders, or other decision makers, are not educated in what the information system can provide them, how the system generates information, and what the representations mean, they may not want to rely on the system to aid in decision making. Erosion of trust can quickly filter down through all levels of the organization and can result in misuse, disuse and ultimate failure of the system.

Technology Requires People. Technology and automation are frequently advanced as work savers. New technology can result in better products, quicker processing, and larger volumes of completed work. However, in providing these improvements, technology frequently results in a redistribution of work rather than a work savings. The 4950th Test Wing provides an excellent example. An initial assumption of the wing was to complete massive technological advances with no increase in personnel or in personal qualifications. However, the wing readily admits that the MIS will require much more data than is currently being saved manually. Someone will have to gather and enter the data into the MIS data base and someone else will have to train the entry personnel to insure the data is stored correctly. If an error is made in manual data collection, anyone with a pencil and eraser can make the correction. If an error is made in the MIS, however, someone with knowledge of the data base structure and command language will have to make the correction. With manual data collection, if the managers want to see data presented in a new format, a typist can generally respond. In the MIS, new formats may require technical experts to reprogram the computer to respond in the desired manner. In sum, advances in technology are not free: they require redirections in the qualifications of the people. For the 4950th Test Wing, this means identifying and grooming data base specialists and overseers.

# BIBLIOGRAPHY


Conway, Richard W., B. M. Johnson, and W. L. Maxwell. "An Experimental Investigation of Priority Dispatching," Journal of Industrial Engineering, 11 (3): 221-229 (May-June 1960).

----- and -----. "Priority Dispatching and Work-In-Progress Inventory," Journal of Industrial Engineering, 16 (2): 123-130 (March-April 1964).

----- and -----. "Priority Dispatcing and Job Lateness in a Job Shop," Journal of Industrial Engineering, 16 (4): 228-237 (July-August 1964).

Cooper, Dale F. "Heuristics for Scheduling Resource-Constrained Projects: An Experimental Investigation," Management Science, 22 (11): 1186-1194 (July, 1976).

Dale, Ernest. Management: Theory and Practice. New York: McGraw-Hill Book Company, 1965).

Davis, E. W. "Project Scheduling Under Resource Constraints: Historical Review and Categorisation of procedures, AIIE Transaction, 5 (1973).

DeSanctis, Gerardine. "Computer Graphics as Decision Aids: Directions for Research," Decision Sciences, 15 (4): 463-483 (Fall, 1984).

Dumond, John. Evaluation of Due Date Assignment Rules for Dynamically Arriving Projects. PhD Dissertation. School of Business, Indiana University, February 1985.

El Sawy, Omar A. "Implementation by Cultural Infusion: An Approach for Managing the Introduction of Information Technology in Organizations," MIS Quarterly 9 (2): 131-140. (June 1985)

Filley, Richard D. "1986 Project Management Software Buyer's Guide," IE, 51-63, 86 (January 1986).

Gavett, J. William, Production and Operations Management. New York: Harcourt, Brace and World, 1968.

Glenn, Col Joseph K. 4950th Test Wing Business Systems Planning Study. Unpublished report. 4950th Test Wing, Wright-Patterson AFB, Ohio, June 1984.

Goodman, David A. "A Goal Programming Approach to Aggregate Planning of Production and Workforce," Management Science, 20 (12): 1569-1575 (1974).

Hillier, Frederick S., and Gerald J. Lieberman. Introduction to Operations Research (Third Edition). Oakland, California: Holden-Day, Inc., 1980.

IBM, Business Systems Planning: Information Systems Planning Guide (GE20-0527-3), White Plains, New York: International Business Machines, Corp. 1981

Interviews, Test Directors and Shop Personnel. Personal interviews. 4950th Test Wing, Aeronautical Systems Division, Air Force Systems Command, Wright-Pattersn AFB, Ohio. May - August, 1985.

Keen, Peter G. W. and Michael S. Scott Morton. Decision Support Systems: An Organizational Perspective. Reading, Mass: Addison-Wesley Publishing Company, 1978.

Lee, Sang M. Goal Programming for Decision Analysis. Philadelphia: Auerbach Publishers, Inc., 1972.

Lin, W. T. "A Survey of Goal Programming Applications," Omega, 8 (1): 115-117 (1980).

McGough, Elise H. "Scheduling: Effective Methods and Techniques," American Society of Civil Engineering Proceedings, 108 (CO1, #16900), March 1982.

Meyer, Ken and Mike Harper. "Issues and Opinions," MIS Quarterly 8 (1): 1-2. (March 1984)

Morris, D. J. Communications for Command and Control Systems. Oxford: Pergamon Press, 1983.

Patterson, James H. Exact and Heuristic Solution Procedures for the Contrained-Resource, Project Scheduling Problem. Monograph. November, 1982

Rogers, Captain Steven K. Lecture materials distributed in EENG 5.72, Communications for Command and Control. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, July-August 1985.

Sauder, Richard L. and William M Westerman. "Computer Aided Train Dispatching: Decision Support Through Optimization," Interfaces 13 (6): 24-37. (December, 1983)

Simon, Herbert A. The Science of the Artificial (Second Edition). Cambridge, MA: The MIT Press, 1982.

-----. The New Science of Management Decision. New York: Harper and Row Publishers, 1960.

Sprague, Ralph H. Jr., and Eric D. Carlson. Building Effective Decision Support Systems. Englewood Cliffs, New Jersey: Prentice-Hall Inc, 1982.

Sutton, Lt Col Donald A., Executive Officer. Personal interviews. 4950th Test Wing, Wright-Patterson AFB, Ohio. January through April 1985.

Test Guide. Test Director's Guide, (Third Edition). 4950th TESTWP 80-1, 4950th Test Wing, Aeronautical Systems Division, Air Force Systems Command, Wright-Patterson AFB, Ohio: January 1983.

Test Wing. 4950th Test Wing Information System Tactical Planning Requirements Document. Draft of unpublished report. 4950th Test Wing, Wright-Patterson AFB, Ohio, April 1985.

Young, Lawrence F. "Right Brained DSS," Data Base, 4: 28-36 (Summer 1983).

Valusek, Major John R. Personal interviews and lecture material presented in OPER 6.52, Decision Support Systems. School of Engineering, Air Force Institute of Techology (AU), Wright-Patterson AFB, Ohio, October-December 1985.

Zeleny, Milan. Multiple Criteria Decision Making. New York: McGraw-Hill, 1982.

## Vita

Major Robert H. Black was born on 25 March 1948 in DeLand, Florida. He graduated from DeLand High School in 1966 and received a Bachelor of Science in Geography degree from the United States Air Force Academy in June, 1971. Following pilot training at Craig AFB, AL, he was assigned to Charleston AFB, SC, where he served as a C-141 instructor pilot for the Forty-First Military Airlift Squadron. While at Charleston, he earned a Master of Science in Education degree from Southern Illinois University. Subsequent assignments include The Office of the Chief of Air Force Reserve (The Pentagon), Airmanship Division (United States Air Force Academy), Twenty-Second Air Force (Travis AFB, CA), and Headquarters, Military Airlift Command (Scott AFB, IL). Major Black entered the School of Engineerring, Air Force Institute of Techhnology, in August 1984.

Permanent address: 4 Park Avenue, Hidden Valley
DeLeon Springs, Florida 32028

# Vita

Captain Mark J. Fowler was born on 14 September 1955, in Santa Barbara, California. He graduated from Santa Barbara High School in 1973 and received a Bachelor of Science degree from the United States Air Force Academy in June 1977. Following pilot training at Fort Rucker, AL, he was assigned to Minot AFB, ND, where he served as an H-1 instructor pilot for Detachment 7, 37 Aerospace Rescue and Recovery Squadron (ARRS). He was assigned to F. E. Warren AFB, WY in 1981, where he served as the Chief of Safety for the 37 ARRS. Captain Fowler entered the School of Engineering, Air Force Institute of Technology, in August 1984.

Permanent address: 830 Eucalyptus Ave.
Santa Barbara, California 93101

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT<br>Approved for Public Release;<br>distribution unlimited |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S)<br>AFIT/GST/ENS/86M-2 | 5. MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a. NAME OF PERFORMING ORGANIZATION<br>School of Engineering — 6b. OFFICE SYMBOL (If applicable)<br>AFIT/ENS | 7a. NAME OF MONITORING ORGANIZATION |
| 6c. ADDRESS (City, State and ZIP Code)<br>Air Force Institute of Technology<br>Wright-Patterson AFB, OH 45433 | 7b. ADDRESS (City, State and ZIP Code) |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION<br>4950th Test Wing (AFSC) — 8b. OFFICE SYMBOL (If applicable) | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
| 8c. ADDRESS (City, State and ZIP Code) | 10. SOURCE OF FUNDING NOS.<br>PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT NO. |
| 11. TITLE (Include Security Classification)<br>See Box 19 | |

12. PERSONAL AUTHOR(S)
ROBERT H. BLACK, B.S., M.S., MAJ, USAF MARK J. FOWLER, B.S., CAPT, USAF

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Yr., Mo., Day) | 15. PAGE COUNT |
|---|---|---|---|
| MS Thesis | FROM ______ TO ______ | 1986 March | 100 |

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

| FIELD | GROUP | SUB. GR. |
|---|---|---|
| 05 | 10 | |
| 12 | 2 | |

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Decision Support Systems, Systems Analysis, Project Scheduling, Project Management

19. ABSTRACT (Continue on reverse if necessary and identify by block number)


Title: INVESTIGATION AND DESIGN OF A PROJECT MANAGEMENT DECISION SUPPORT SYSTEM FOR THE 4950TH TEST WING

Thesis Chairman: John R. Valusek, MAJ, USAF
Assistant Professor of Operations Research


Approved for public release: IAW AFR 190-1.
Lynn Wolaver 9 May 86
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

| | |
|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐ | 21. ABSTRACT SECURITY CLASSIFICATION<br>UNCLASSIFIED |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL<br>John R. Valusek, MAJ, USAF | 22b. TELEPHONE NUMBER (Include Area Code) (513) 255-3362 — 22c. OFFICE SYMBOL AFIT/ENS |

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

END

10-86

DTIC